

UNIVERSIDAD NACIONAL DE PIURA

Facultad de Ingeniería Industrial

Departamento de Ingeniería Informática



“ MÓDULO DE MODELADO DE DATOS”

Ing°. JORGE ALVARADO TABACCHI

INDICE

CAPÍTULO 1. MODELO DE DATOS	5
1.1. Introducción al Modelo de Datos.....	5
1.2. Modelo, Esquema y Ejemplar	5
1.3. Tipos de Abstracción en el Diseño de base de datos	6
1.3.1. Clasificación/Particularización	6
1.3.2. Agregación/Desagregación	6
1.3.3. Generalización/Especialización	7
1.3.4. Jerarquías de Abstracción	8
1.4. Concepto de Modelo de Datos	8
1.4.1. Estática	8
1.4.2. Dinámica.....	9
1.5. Componentes de una restricción	10
CAPÍTULO 2. MODELO ENTIDAD/INTERRELACION (MER)	11
2.1. Presentación del Modelo	11
2.2. Estática del Modelo E/R	11
2.2.1. Entidad	11
2.2.2. Interrelación	12
2.2.3. Dominio y valor	12
2.2.4. Atributo	13
2.3. Restricciones.....	14
2.4. Semántica de las Interrelaciones.....	15
2.4.1. Elementos de un tipo de interrelación.....	15
2.4.2. Cardinalidad de un tipo de entidad.....	16
2.4.3. Atributos de las interrelaciones	17
2.4.4. Dependencia en existencia y en identificación	17
2.5. Control de Redundancia	18
2.5.1. Atributos derivados.....	18
2.5.2. Interrelaciones redundantes	18
2.6. Interrelaciones de grado superior a 2	19
2.6. Otras restricciones sobre interrelaciones	20
2.6.1. Restricción de Exclusividad	20
2.6.2. Restricción de exclusión	21
2.6.3. Restricción de Inclusividad	22

2.6.4.	Restricción de Inclusión.....	22
2.7.	Generalización/Especialización	22
2.8.	Cómo desarrollar un Esquema Entidad-Interrelación	23
CAPÍTULO 3. MODELO DE DATOS RELACIONAL		31
3.1.	Presentación del Modelo	31
3.2.	Elementos permitidos	31
3.2.1.	Dominios, Relaciones y Atributos.....	31
3.3.	Intensión y Extensión de una Relación.....	32
3.4.	Elementos no permitidos: Restricciones	32
3.4.1.	Restricciones inherentes	32
3.4.2.	Restricciones semánticas	32
3.4.2.1.	Clasificación según los elementos a los que afecta la condición	33
3.4.2.2.	Restricciones de condición y acción específicas	33
3.4.2.3.	Restricciones de Condición General (Predicado Libre) y de Acción Específica (Rechazo) 35	
3.4.2.4.	Restricciones de Condición General y Acción General (Disparadores)	35
CAPÍTULO 4. DEPENDENCIAS FUNCIONALES		36
4.1.	Dependencias entre los datos	36
4.2.	Concepto de Dependencia Funcional	36
4.2.1.	Dependencia funcional plena o completa.....	37
4.2.2.	Dependencia Funcional trivial.....	38
4.2.3.	Dependencia funcional elemental	38
4.2.4.	Dependencia funcional transitiva.....	38
CAPÍTULO 5. TEORÍA DE LA NORMALIZACIÓN		40
5.1.	Necesidad de un método formal de diseño relacional	40
5.2.	Teoría Formal de la Normalización	41
5.2.1.	Conservación de la información.....	42
5.2.2.	Conservación de las dependencias	42
5.3.	Definición Formal de las tres primeras formas normales.	42
5.3.1.	Primera Forma Normal (1FN)	43
5.3.2.	Segunda Forma Normal (2FN)	43
5.3.3.	Tercera Forma Normal (3FN)	44
5.3.4.	Forma Normal de Boyce-Codd (FNBC)	45

Ejercicios Resueltos	46
Ejercicios Propuestos	49
CAPÍTULO 6. ALGEBRA RELACIONAL Y LENGUAJE SQL	53
6.1. Algebra Relacional	53
6.1.1. Unión	53
6.1.2. Diferencia	53
6.1.3. Producto Cartesiano	54
6.1.4. Selección	54
6.1.5. Proyección	54
6.1.6. Intersección	54
6.1.7. Join	55
6.1.7.1. Join Natural	55
6.1.7.2. Outer Join	55
6.2. Lenguaje SQL	55
6.2.1. Comandos SQL	56
6.2.2. Cláusulas	56
Ejercicios Resueltos	57
Ejercicios Propuestos	58
BIBLIOGRAFÍA	60

CAPÍTULO 1. MODELO DE DATOS

1.1. Introducción al Modelo de Datos

Los datos siempre han sido registrados en algún tipo de soporte (papel, piedra, madera). Cuando los datos se procesan y se les incorpora un significado entonces hablamos de información. Los datos por sí solos no tienen utilidad sino se les incorpora significado, por ello surgen los modelos de datos que permiten lograr este objetivo.

Un modelo de datos se puede describir como una representación abstracta de los datos de una organización y las relaciones entre ellos. La abstracción es un proceso mental que ayuda a ocultar detalles y fijarse en lo esencial de un objeto, reduciendo así la complejidad y ayudando a la comprensión del mundo real.

Por ejemplo, si queremos modelar en un sistema académico el elemento Alumno, sólo nos fijaríamos en las propiedades esenciales para ese mundo académico, como el código del alumno, su dni, nombre, etc., pero sin tomar en cuenta datos que no nos interesan, como por ejemplo su estatura, alergias, etc.

Los modelos de datos nos proporcionan los mecanismos que permiten la representación de aquella parcela del mundo real cuyos datos nos interesa registrar, lo que habitualmente se denomina universo del discurso. Esta representación se concibe en dos niveles: el de las estructuras (esquema) y el de la información en sí misma (base de datos).

Los modelos de datos también cuentan con lenguajes de datos que permiten definir y manipular la base de datos. Un Lenguaje de Datos es un modelo de datos más una sintaxis:

$$\text{LD} = \text{MD} + \text{Sintaxis}$$

1.2. Modelo, Esquema y Ejemplar

Un Modelo de datos es un instrumento de descripción y el resultado de la misma es el esquema. La descripción de un cierto mundo real por medio de un modelo de datos da como resultado un esquema. También se debe distinguir entre el esquema como la descripción de la estructura de la base de datos y el ejemplar del esquema, que son los datos que en un determinado momento se encuentran almacenados en el esquema. Ejemplo de Esquema:

CURSO	Docente	Alumno
Codigo	Codigo	Codigo
Nombre	Nombre	Nombre
Crédito	DNI	DNI

Ejemplos de Ejemplar:

De Curso	De Docente	De Alumno
001	0001	10201300001
Análisis de Sistema	Oscar Reyes	Miguel Miranda
4	02658476	72458967
002	0002	102013000002
Modelamiento de BD	Pedro Criollo	Ana Rojas
4	02648756	02654873

El esquema en principio, es invariante en el tiempo (estático), sólo puede cambiar si es que se produce un cambio en el mundo real o en nuestra percepción del mismo. En cambio, los ejemplares pueden variar en el transcurso del tiempo, debido a registros nuevos, modificaciones, o borrados, se dice entonces que son dinámicos.

1.3. Tipos de Abstracción en el Diseño de base de datos

Los modelos de datos ofrecen distintos tipos de abstracciones a fin de facilitar la representación de los datos, siendo el esquema el resultado de aplicar un proceso de abstracción a un determinado mundo real. Suelen ser de cuatro tipos: Clasificación, Agregación, Generalización y Asociación.

1.3.1. Clasificación/Particularización

La clasificación es la acción de abstraer las características comunes a un conjunto de ejemplares para crear una categoría a la cual pertenecen dichos ejemplares. El proceso inverso de la clasificación es la particularización que consiste en pasar de la clase a sus ejemplares. Ver la figura 1.1

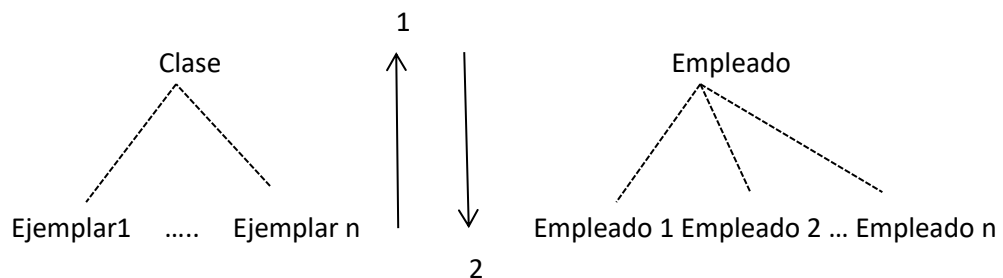


Figura 1.1. Clasificación (1) y Particularización (2)

1.3.2. Agregación/Desagregación

La abstracción de agregación consiste en construir un nuevo elemento del modelo como compuesto de otros elementos (componentes). Se pueden considerar tres tipos de agregación:

- **Agregación de clases para obtener una clase compuesta.** Por ejemplo, FACULTAD, se puede obtener mediante una abstracción de

agregación a partir de la clase ESCUELA, considerándolo un conjunto de diferentes escuelas.

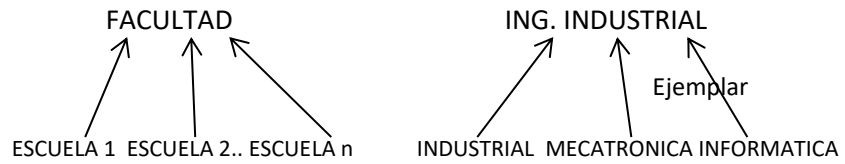


Figura 1.2 Ejemplo de agregación de clases y un ejemplar

- **Agregación de propiedades para obtener una clase.** Po ejemplo, DOCENTE se puede considerar como agregación de sus propiedades DNI, Nombre, Dirección.

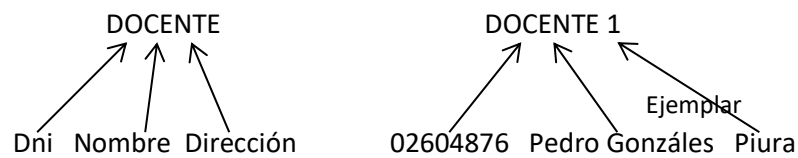


Figura 1.3 Ejemplo de agregación de propiedades y un ejemplar

- **Agregación de propiedades para obtener una propiedad compuesta.** Por ejemplo la agregación de Día, Mes y Año para obtener una Fecha.

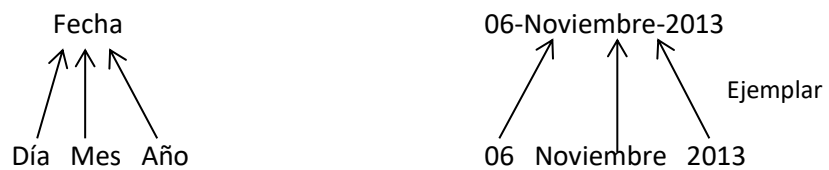


Figura 1.4 Ejemplo de agregación de propiedades para obtener una propiedad compuesta y un ejemplar

El proceso inverso a la agregación es la desagregación.

1.3.3. Generalización/Especialización

La generalización es la acción de abstraer las características comunes a varias clases (subclases) para construir una clase más general (superclase). El proceso inverso de la generalización es la especialización.

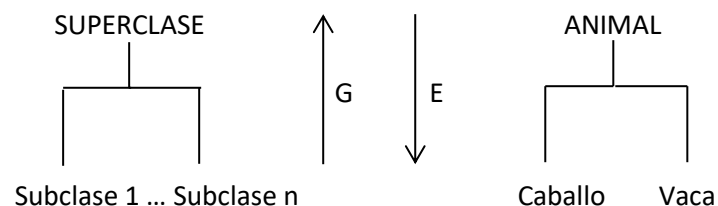


Figura 1.5 Representación y ejemplo de Generalización (G) y Especialización (E)

1.3.4. Jerarquías de Abstracción

Es posible combinar distintas abstracciones, formando lo que se conoce como jerarquía de abstracciones. En la siguiente figura se muestra un ejemplo de ello.

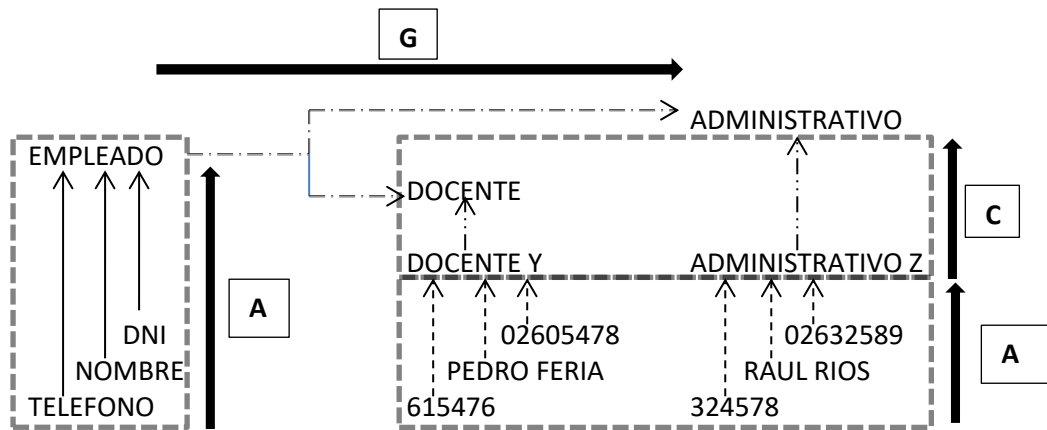


Figura 1.6. Jerarquía de Abstracción. G: Generalización, A: Agregación, C: Clasificación

1.4. Concepto de Modelo de Datos

Un modelo de datos se define como un conjunto de conceptos, reglas y convenciones bien definidos que nos permiten aplicar una serie de abstracciones a fin de describir y manipular los datos de un cierto mundo real que deseamos almacenar en la base de datos.

También se puede afirmar que las propiedades del mundo real son de dos tipos:

- Estáticas, relativamente invariantes en el tiempo, que responden a lo que se suele entender como estructuras.
- Dinámicas, que son las operaciones que se aplican a los datos o valores almacenados en las estructuras, los cuales varían en el transcurso del tiempo al aplicársele dichas operaciones.

La componente estática de un determinado modelo de datos expresado en una sintaxis es el Lenguaje de Definición de Datos (LDD), y la componente dinámica el Lenguaje de Manipulación de Datos (LMD); ambos constituyen el Lenguaje de Datos.

1.4.1. Estática

La estática de un modelo de datos está compuesta por:

1. Elementos permitidos. En general son:
 - 1.1. Objetos (entidades, relaciones, registros, etc.)
 - 1.2. Asociaciones entre objetos (interrelaciones)
 - 1.3. Propiedades o características de los objetos o asociaciones (atributos, campos, elementos de datos, etc.).
 - 1.4. Dominios, que son conjuntos nominados de valores homogéneos sobre los que se definen las propiedades.

2. Elementos no permitidos o restricciones. Existen limitaciones que unas veces son impuestas por el modelo de datos y otras las impone el mismo universo del discurso y se denominan restricciones; las impuestas por el modelo son restricciones inherentes y las que se ajustan a la representación más exacta del mundo real son restricciones de integridad o semánticas.
 - 2.1. Restricciones inherentes: Son aquellas impuestas por el mismo modelo de datos, al no admitir ciertas estructuras introduce ciertas rigideces al momento. El usuario (diseñador) no controla estas restricciones.
 - 2.2. Restricciones de integridad o semánticas, son aquellas que permiten captar la semántica del universo del discurso que se quiere modelar. El usuario (diseñador) ha de definir, y a veces programar, estas restricciones a fin de limitar los valores que pueden tomar los datos de la base de datos o de impedir ciertos cambios de los mismos. Las restricciones semánticas pueden ser:
 - 2.2.1. Reconocidas por el MD. Su definición le corresponde al diseñador, pero su gestión es responsabilidad del modelo de datos, el cual las reconoce y recoge en el esquema, suministrando instrumentos para su definición y obligando a su cumplimiento.
 - 2.2.2. Ajenas al MD. Son completamente responsabilidad del diseñador ya que el modelo de datos no las reconoce ni proporciona instrumentos para manejarlas.

1.4.2. *Dinámica*

El conjunto de valores que toman las distintas categorías de un esquema en un momento determinado t_i recibe el nombre de ejemplar de esquema o estado de la base de datos en el tiempo t_i (BD_i) en otro momento t_j el ejemplar del esquema será BD_j . Si entre t_i y t_j se ha producido un cambio en algún valor de la base de datos (alta, baja o modificación) $BD_i \neq BD_j$. Tanto BD_i como BD_j deben ser ejemplares válidos de la base de datos.

La componente dinámica del modelo consta de un conjunto de operadores que se definen sobre la estructura del correspondiente modelo de datos, ya que no todas las estructuras admiten el mismo tipo de operaciones. La aplicación de un operador a un ejemplar de un esquema transforma éste en otro ejemplar:

$$O[BD_i] = BD_j$$

Pudiendo ser $BD_i = BD_j$, por ejemplo en caso de consulta o cuando falla una operación por haberse producido un error.

Una operación tiene dos componentes:

1. Localización, consiste en localizar un ejemplar de un objeto indicando un camino, o un conjunto de ejemplares especificando una condición.

2. Acción, que se realiza sobre los ejemplares previamente localizados mediante una operación de localización y puede consistir en una recuperación o en una actualización.

1.5. Componentes de una restricción

En una restricción de integridad es posible distinguir los siguientes componentes:

- La operación de actualización (inserción, borrado o modificación) cuya ejecución ha de dar lugar a la comprobación del cumplimiento de la restricción.
- La condición que debe cumplirse, la cual es en general una proposición lógica definida sobre uno o varios elementos del esquema, que puede tomar uno de los valores de verdad (verdadero o falso).
- La acción que debe llevarse a cabo dependiendo del resultado de evaluar la condición.

Las restricciones de integridad se pueden considerar, en cierto modo, como reglas ECA (Evento, Condición, Acción), en las cuales, al ocurrir un evento (en este caso una actualización), se comprueba una condición y dependiendo de su resultado se pone en marcha una acción (rechazar la operación, informar al usuario, corregir el error, etc.)

Las restricciones han de ser definidas en la fase de diseño y el cumplimiento de la condición tiene que ser verificado en la de ejecución.

CAPÍTULO 2. MODELO ENTIDAD/INTERRELACION (MER)

2.1. Presentación del Modelo

El Modelo Entidad Interrelación (E/R) fue propuesto por Peter Chen en 1976 que proponía una vista unificada de los datos, adoptando el enfoque más natural del mundo real que consiste en entidades e interrelaciones.

Posteriormente otros muchos autores, como Teorey, FARG, Elsmari, propusieron importantes aportaciones a la propuesta inicial de Chen, con lo que se conoce como modelo E/R extendido.

El modelo E/R es un modelo conceptual que permite concebir la base de datos a un nivel superior de abstracción, aislándolo de consideraciones relativas a la máquina (tanto en su nivel lógico como físico).

2.2. Estática del Modelo E/R

En el modelo E/R propuesto por Chen, se distinguen los siguientes elementos: Entidad, Interrelación, Atributo y Dominio.

2.2.1. Entidad

Se define como cualquier objeto (real o abstracto) que existe en la realidad y acerca del cual queremos almacenar información. La estructura genérica que describe un conjunto de entidades aplicando la abstracción de clasificación se denomina tipo de entidad, mientras que entidad es cada uno de los ejemplares de ese tipo de entidad.

El conjunto de ejemplares de un tipo de entidad en un momento dado será la extensión de ese tipo de entidad, mientras que la intensión es el tipo de entidad propiamente dicho. Matemáticamente, un conjunto de ejemplares de un tipo de entidad se define como:

$$\{ e: p(E) \}$$

Siendo e un ejemplar del tipo de entidad E y p el predicado asociado a E . Por ejemplo, el tipo de entidad ALUMNO, cuyo predicado asociado es “Persona que realiza estudios”, tiene un ejemplar “Feria” que pertenece a él, ya que cumple dicho predicado.

La representación gráfica de un tipo de entidad es un rectángulo etiquetado, en cuyo interior está el nombre del tipo de entidad, como vemos en la figura 2.1.



Figura 2.1. Representación de tipos de entidad

Existen dos clases de entidades: regulares o fuertes, que son aquellas cuyos ejemplares tiene existencia por sí mismos (como ALUMNO y DOCENTE), y débiles en las cuales la existencia de un ejemplar depende de que exista cierto ejemplar de otro tipo de entidad, por ejemplo INSCRIPCION depende de CURSO, y la desaparición de un determinado curso hace que también desaparezcan todas las inscripciones de curso.

Los tipos de entidad débil se representan con dos rectángulos concéntricos con su nombre en el interior, como se puede observar en la figura 2.2.

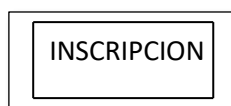


Figura 2.2. Representación de una entidad débil

2.2.2. Interrelación

Se entiende por interrelación una asociación, vinculación o correspondencia entre entidades. También se denomina tipo de interrelación a la estructura genérica que describe un conjunto de interrelaciones, mientras que la interrelación será cada uno de los ejemplares concretos. Por ejemplo, **Habita** es un tipo de interrelación que vincula los tipos de entidad PERSONA y VIVIENDA.

Matemáticamente, el conjunto de interrelaciones de un tipo de interrelación I se define como:

$$\{ \langle e_1, e_2, \dots, e_n \rangle \}$$

Donde e_i es un ejemplar de un tipo de entidad E_i y n el grado de tipo de interrelación, es decir el número de tipos de entidad que están asociados en el tipo de interrelación.

El tipo de interrelación se representa mediante un rombo etiquetado con el nombre de la interrelación, unido mediante arcos a los tipos de entidad que asocia, como se puede observar en la figura 2.3.

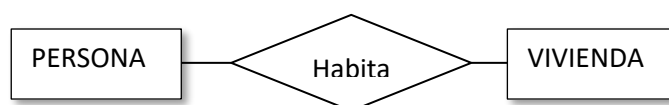


Figura 2.3. Representación de la Interrelación Habita

2.2.3. Dominio y valor

El conjunto de posibles valores que puede tomar una cierta característica de un tipo de entidad se denomina **dominio**. Se define el dominio como un conjunto de valores homogéneos con un nombre. Para saber si un valor pertenece a un dominio determinado, comprobaremos que cumple el predicado que el dominio lleva siempre asociado. Matemáticamente se expresa:

$$D = \{ v_i : p(v_i) \}$$

Donde D es el dominio, v_i es un valor y p es el predicado asociado a dicho dominio.

Por ejemplo, el valor “uno” se toma del dominio Números, y cumple el predicado de ser uno de los elementos posibles del conjunto (“uno”, “dos”, “tres”). Un dominio se puede definir por intensión, especificando el tipo de datos (por ejemplo, carácter 50 para el nombre de alumno); o por extensión, declarando el valor de cada elemento del dominio (como el caso de Números). El dominio se representa gráficamente con un círculo u óvalo etiquetado con su nombre. En la figura 2.4 se muestran dos formas de representar un dominio.



Figura 2.4. Dos formas de representar un dominio

2.2.4. Atributo

Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de interrelación se denomina atributo; los atributos toman valores de uno o varios dominios. Matemáticamente consiste en una función de un tipo de entidad o de interrelación sobre todos los posibles subconjuntos de los valores de un dominio o de un conjunto de dominios:

$$A: E \rightarrow S(D) \text{ ó } A: E \rightarrow S(D_1) \times S(D_2) \times \dots \times S(D_n)$$

$$A: I \rightarrow S(D) \text{ ó } A: I \rightarrow S(D_1) \times S(D_2) \times \dots \times S(D_n)$$

Donde A es el atributo, $S(D_n)$ todos los posibles subconjuntos de los valores de los dominios, E es el tipo de entidad e I el tipo de interrelación.

La representación gráfica de un atributo consiste en cualificar con su nombre el arco que une el dominio con el tipo de entidad o de interrelación (ver figura 2.5). Sin embargo, para simplificar la representación gráfica y siempre que coincida el nombre del dominio con el atributo, será suficiente con el círculo u óvalo con el nombre del atributo.

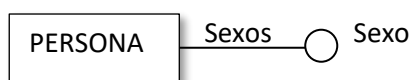
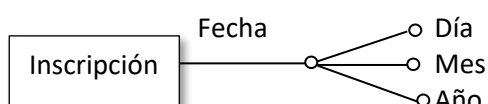


Figura 2.5. Representación de dominio y de atributo

El modelo E/R admite atributos compuestos, es decir, atributos definidos sobre más de un dominio. Por ejemplo, la fecha de inscripción puede estar definido sobre los dominios Día, Mes, Año. En la figura 2.6 se muestran dos formas de representar los atributos compuestos.



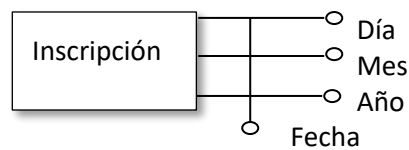


Figura 2.6. Representación de atributos compuestos

2.3. Restricciones

El modelo E/R tiene como restricción inherente que sólo permite establecer interrelaciones entre entidades, no estando admitidas entre entidades e interrelaciones ni entre interrelaciones. También obliga el modelo a que todas las entidades tengan un identificador, lo que asimismo podría considerarse una restricción inherente. El no tener apenas restricciones inherentes dota al modelo de un gran flexibilidad para la representación del mundo real.

En cuanto a las restricciones de integridad, existen las restricciones sobre valores y las estructurales.

Las restricciones sobre los valores se establecen mediante la definición del dominio, la cual permite limitar los valores del dominio y, por tanto, los de los atributos sobre él definidos, a los de un determinado tipo de datos, o restringirlos a los comprendidos en un rango, o bien declarar los valores posibles en el caso de que la definición se haga por extensión.

Las restricciones estructurales se refieren tanto a atributos como interrelaciones.

De las referidas a los atributos existe aquella en la cual de entre todos los atributos de un tipo de entidad han de existir uno o varios que identifiquen unívocamente cada uno de los ejemplares de este tipo de entidad. Cada uno de estos conjuntos de atributos se denomina Identificador Candidato (IC). Todo IC debe cumplir la condición de ser unívoco y mínimo. Entre los IC se elige uno como Identificador Principal (IP) y el resto serán Identificadores Alternativos (IA). Su representación gráfica se presenta en la figura 2.7.

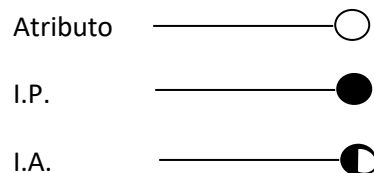


Figura 2.7. Representación gráfica de IP e IA

También están permitidos en el modelo E/R los atributos multivaluados y opcionales (nulos o "faltantes"). Los atributos multivaluados son aquellos que pueden tomar más de un valor del dominio subyacente frente a los univaluados que toman un solo valor. También se puede obligar a que un atributo tome como mínimo un valor del dominio subyacente, es decir el valor de ese atributo es obligatorio (no puede ser nulo). En la

figura 2.8 se ve la representación de los atributos multivaluados/univaluados y opcionales/obligatorios.



Figura 2.8. Ejemplos de atributos multivaluado y opcional

Estas restricciones pueden ser definidas sobre el concepto de cardinalidad de un atributo. Se entiende por cardinalidad mínima (o máxima) de un atributo al número mínimo (o máximo) de valores que puede tomar ese atributo en cada ejemplar del tipo de entidad al cual pertenece. Las cardinalidades se representan asociando un par de números enteros (mín, máx) al correspondiente atributo. En la figura 2.9 aparecen los cuatro tipos de cardinalidades.

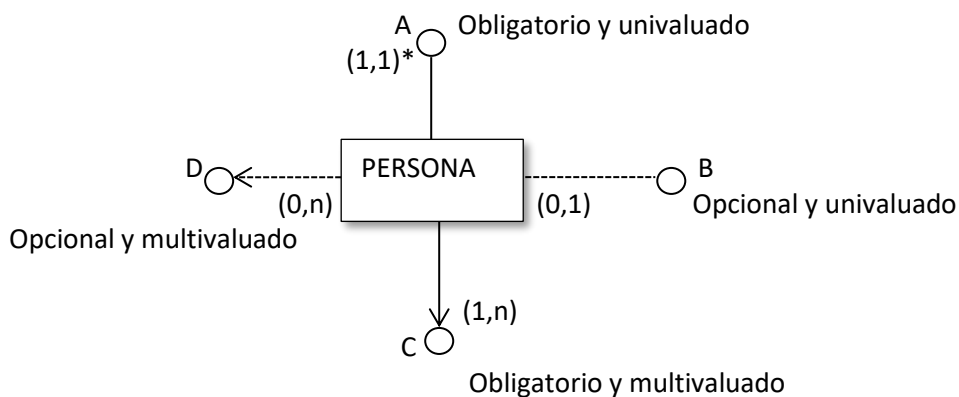


Figura 2.9. Representación de los cuatro tipos posibles de cardinalidades

2.4. Semántica de las Interrelaciones

2.4.1. Elementos de un tipo de interrelación

En un tipo de interrelación se distinguen los siguientes elementos:

- **Nombre:** Cada tipo de entidad tiene un nombre único que lo distingue del resto.
- **Grado:** Es el número de tipos de entidad que participan en un tipo de interrelación. Cuando un tipo de interrelación asocia dos tipos de entidad se dice que es de grado 2 o binaria, son los tipos de interrelación más comunes. Un caso particular de interrelaciones binarias son las reflexivas (o recursivas), las cuales asocian un tipo de entidad consigo misma. En la figura 2.10 se observa una interrelación recursiva Contiene, en la cual se aprecia que una página Web puede a su vez contener otras páginas web.

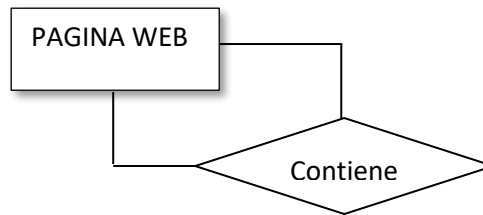


Figura 2.10 Ejemplo de tipo de interrelación reflexiva

- **Tipo de correspondencia:** Es el número máximo de ejemplares de un tipo de entidad que pueden estar asociados, en una determinada interrelación, con un ejemplar de otro(s) tipo(s). Su representación gráfica se hace poniendo una etiqueta con 1:1, 1:N o N:M, según corresponda el lado de la interrelación, también se representa poniendo una flecha del lado de N. En la figura 2.11 se ven ejemplos con ambas representaciones.

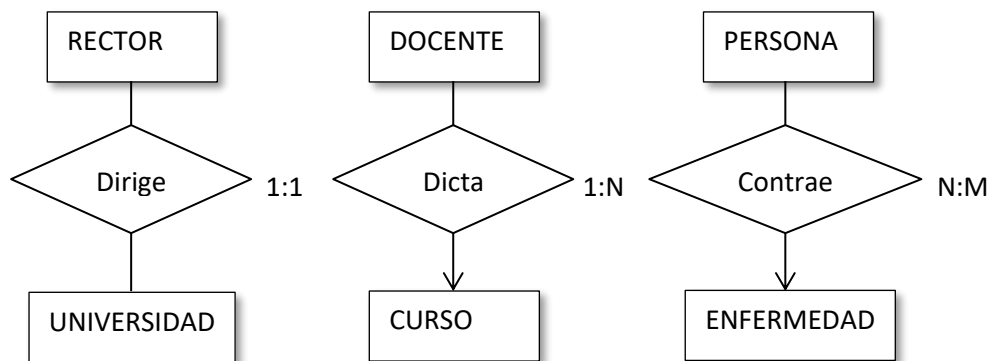


Figura 2.12. Ejemplos de tipo de correspondencia

- **Papel ("rol"):** Es la función que cada uno de los tipos de entidad realiza en el tipo de interrelación. Siempre que no exista ambigüedad se suele prescindir de representar el papel.

2.4.2. Cardinalidad de un tipo de entidad

Se define como el número máximo y mínimo de ejemplares de un tipo de entidad que pueden estar interrelacionadas con un ejemplar del otro, u otros tipos de entidad que participan en el tipo de interrelación. Se representa gráficamente mediante una etiqueta del tipo (0,1), (1,1), (0,n) ó (1,n). Ver figura 2.13.

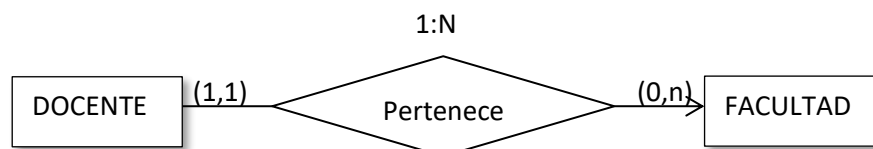


Figura 2.13. Ejemplo de cardinalidades

2.4.3. Atributos de las interrelaciones

Cuando una interrelación 1:N tiene un atributo asociado, se puede demostrar matemáticamente que el atributo puede llevarse a la entidad cuya cardinalidad máxima es N.

Los atributos de las interrelaciones N:M, son propios de la misma y no de las entidades vinculadas por la interrelación.

2.4.4. Dependencia en existencia y en identificación

Los tipos de interrelación también se clasifican en regulares y débiles, según estén asociados dos tipos de entidades regulares, o un tipo de entidad débil con un tipo de entidad regular o débil, respectivamente. En el tipo de interrelación débil se pueden distinguir la dependencia en existencia y la dependencia en identificación. Se dice que hay dependencia en existencia cuando los ejemplares de un tipo de entidad débil no pueden existir si desaparece el ejemplar del tipo de entidad regular del cual dependen. Se dice que existe dependencia en identificación cuando, además de cumplirse la condición anterior, los ejemplares del tipo de entidad débil no se pueden identificar por sí mismos, es decir, mediante los propios atributos del tipo de entidad, y exigen añadir el identificador principal del tipo de entidad regular del cual dependen.

Si existe dependencia en identificación, se etiqueta el rombo de la interrelación con **ID**, y con una E (o sin etiqueta) en caso de que la dependencia sea en existencia.

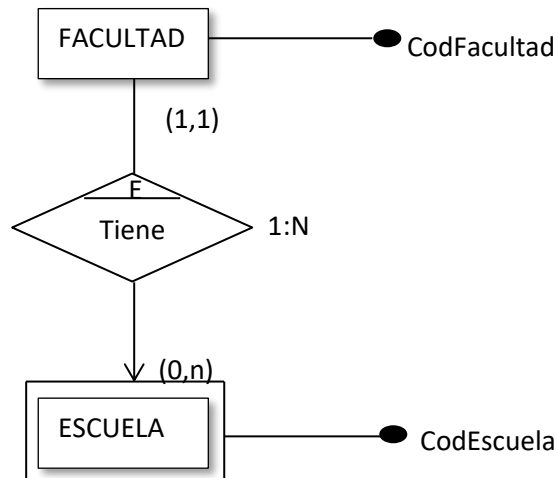


Figura 2.14. Dependencia en existencia

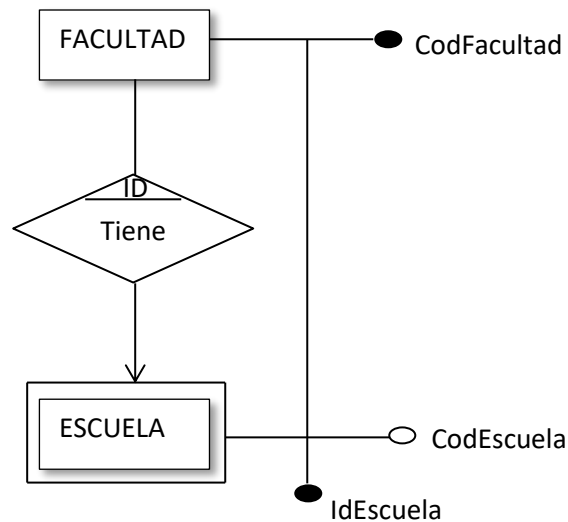


Figura 2.15. Dependencia en Identificación

En la figura 2.14 se representa una dependencia en existencia, donde las escuelas existen mientras exista la facultad a la que pertenecen, sin embargo cada entidad tiene su propio código. Por ejemplo, para la Facultad 01, 02, 03 y para la escuela 001, 002, 003, etc.

En la dependencia por identificación de la Figura 2.15, se aprecia que el IdEscuela depende del CodFacultad, por ejemplo los IdEscuela serían 01001, 01002, 02001, 02002, etc., compuestos por CodFacultad y CodEscuela.

2.5. Control de Redundancia

Es preciso en los esquemas E/R, analizar la existencia de redundancias, por los problemas de inconsistencias a los que pueden dar lugar.

Decimos que un elemento de un esquema es redundante cuando puede ser eliminado sin pérdida de semántica.

Existen dos formas principales de redundancia, según el elemento del modelo E/R al que está asociada: redundancia de los atributos (atributos derivados) y redundancia en las interrelaciones.

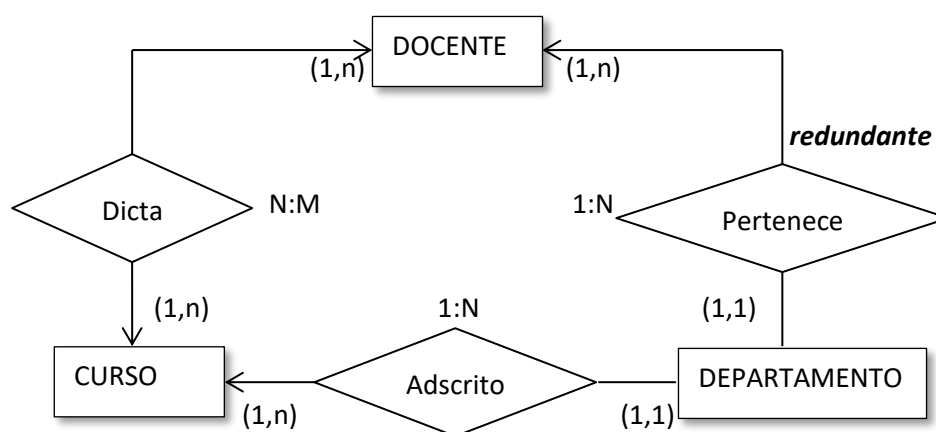
2.5.1. Atributos derivados

Los atributos derivados (o calculados) son aquellos que se obtienen a partir de otros ya existentes, por lo que, aunque son redundantes, no dan lugar a inconsistencia, siempre que en el esquema se indique su condición de derivados y la fórmula mediante la que han de ser calculados. Un ejemplo puede ser el Precio Total de la venta de un producto, el mismo que es calculado multiplicando el precio unitario por la cantidad vendida.

2.5.2. Interrelaciones redundantes

Se dice que una interrelación es redundante cuando su eliminación no implica pérdida de semántica porque existe la posibilidad de realizar la misma asociación de ejemplares por medio de otras interrelaciones.

- a) Que exista un ciclo
- b) Que las interrelaciones que componen el ciclo sean equivalentes semánticamente.
- c) Que se puedan asociar los ejemplares de las dos entidades que estaban interrelacionadas, aun habiéndose eliminado la interrelación.
- d) Que la interrelación o bien no tenga atributos o bien éstos puedan ser transferidos a otra a fin de no perder semántica.



En esta figura se tienen los siguientes supuestos semánticos: Un docente sólo puede impartir cursos que estén adscritos al departamento al que él pertenece; en cuyo caso, si se conocen los cursos que dicta un profesor y el departamento al que está adscrito cada curso, se deduce inmediatamente a qué departamento pertenece dicho docente; de forma análoga, dado un departamento, si sabemos qué cursos tienen adscritos y los profesores que imparten dichos cursos, conoceremos qué docentes pertenecen a dicho departamento, por lo que la interrelación **Pertenece** entre las entidades DOCENTE y DEPARTAMENTO es redundante, su eliminación no produce pérdida de información ni de semántica.

2.6. Interrelaciones de grado superior a 2

19

veces esa descomposición no es posible ya que la semántica recogida en una y otra solución no es la misma.

En la figura 2.17 se puede apreciar una interrelación ternaria en la cual la interrelación Publica no puede ser descompuesta con las interrelaciones Investiga y Participa sin perder semántica.

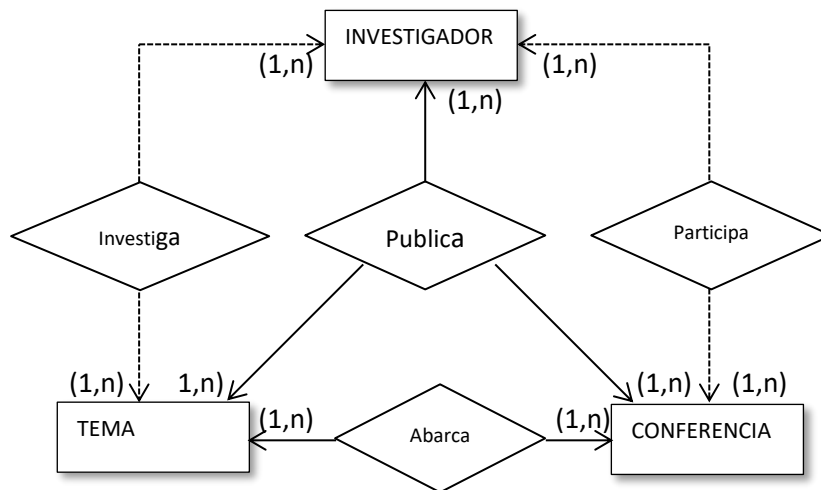


Figura 2.17. Interrelación de grado 3, que no puede ser descompuesta sin pérdida de semántica

2.6. Otras restricciones sobre interrelaciones

2.6.1. Restricción de Exclusividad

Dos o más tipos de interrelación tienen una restricción de exclusividad con respecto a un tipo de entidad que participa en ambas interrelaciones cuando cada ejemplar de dicho tipo de entidad sólo puede pertenecer a uno de los tipos de la interrelación, pero en el momento que pertenezca a uno ya no podrá formar parte del otro. Por ejemplo, en la figura 2.18 se representa una restricción de exclusividad, en la cual si suponemos que un profesor puede dictar cursos de doctorado o recibirlos, pero no ambas cosas, tendríamos una interrelación Imparte y otra Recibe, entre DOCENTE y CURSO, con una restricción de exclusividad entre sí. El arco señala las restricciones que son exclusivas.

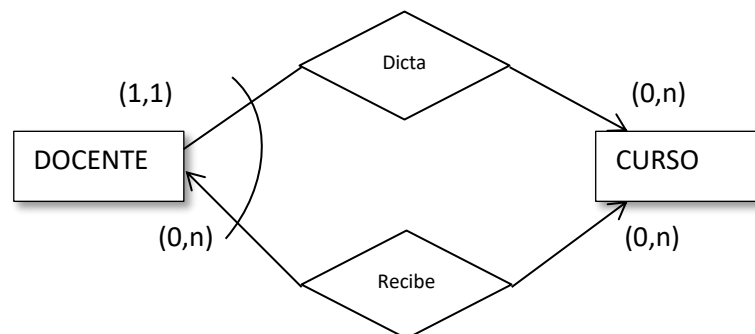


Figura 2.18. Tipo de interrelación exclusiva

En la figura 2.19 se muestra una representación para las interrelaciones exclusivas en la que, además de la cardinalidad de DOCENTE con respecto a **Imparte** y **Recibe**, por separado, se muestra la cardinalidad de DOCENTE con respecto a ambas interrelaciones.

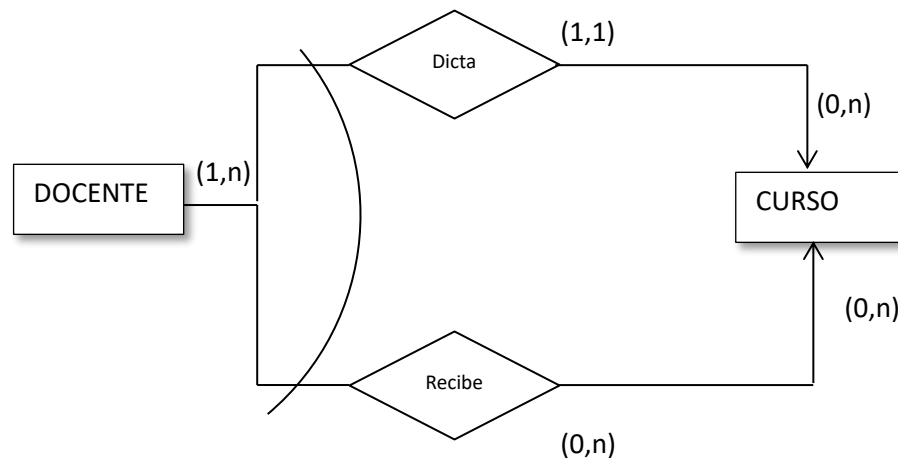


Figura 2.19. Otra representación de una interrelación exclusiva

2.6.2. Restricción de exclusión

Supongamos ahora que se permite a un docente ya doctor matricularse en cursos aunque él, a su vez, esté dictando otros cursos. En este caso la restricción que se impone es que un docente no esté dictando y recibiendo el mismo curso. Es decir, que todo ejemplar de docente que esté unido a un ejemplar de curso mediante la interrelación **imparte**, no podrá estar unido al mismo ejemplar de curso mediante la interrelación **recibe**. En este caso decimos que existe una restricción de exclusión y se representa tal como aparece en la figura 2.20.

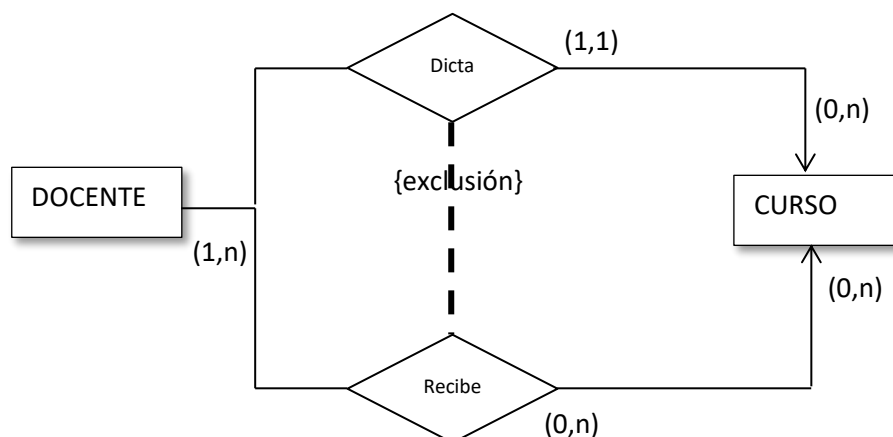


Figura 2.20 Tipo de interrelación con restricción de exclusión

2.6.3. Restricción de Inclusividad

Supongamos ahora que se desea imponer la restricción de que sólo se pueden dictar clases en el programa de doctorado de aquellos docentes que hayan realizado al menos un curso dentro de este mismo programa, aunque no tiene por qué ser el mismo que él imparte. Aplicamos entonces una restricción de inclusividad, por la cual todo ejemplar de un tipo de entidad que participa en uno de los tipos de interrelación tiene necesariamente que participar en la otra. En la figura 2.21 se muestra una representación gráfica de este tipo de interrelación.

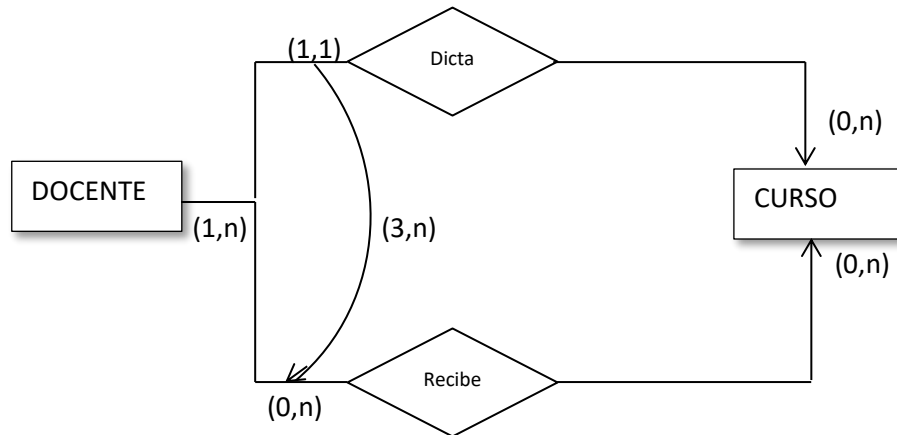


Figura 2.21 Tipo de interrelación con restricción de inclusividad

2.6.4. Restricción de Inclusión

A veces es preciso imponer una restricción más fuerte: si un docente dicta un curso es porque previamente ha tenido que recibir dicho curso. Se aplica entonces una restricción de inclusión como se representa en la figura 2.22.

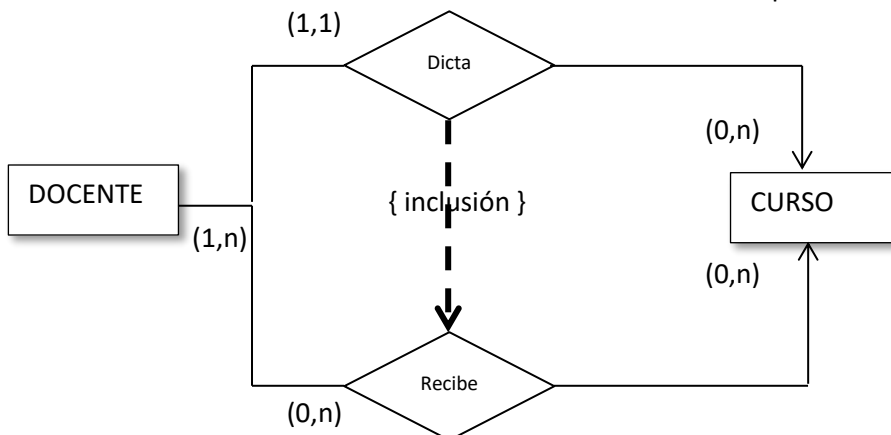


Figura 2.22 Tipo de interrelación con restricción de inclusión

2.7. Generalización/Especialización

Es una abstracción de la extensión del modelo E/R. Se considera como un caso especial de interrelación entre varios tipos de entidad (subtipos) y un tipo más general (supertipo), cuyas características son comunes a todos los subtipos. La interrelación que se establece entre los subtipos y el supertipo corresponde a la noción de “es un”.

Para su representación se utiliza un triángulo cuya base es paralela al rectángulo que representa la entidad del supertipo al cual está conectado, tal como se muestra en la figura 2.23.

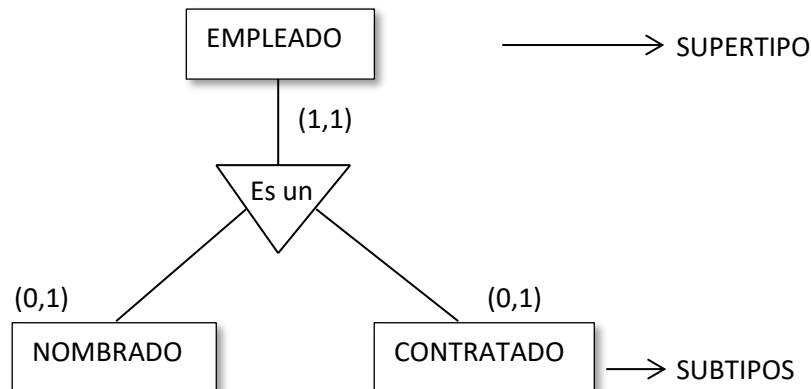


Figura 2.23. Jerarquía de supertipos y subtipos

La abstracción de generalización/especialización tiene las siguientes restricciones semánticas. Atendiendo a si los subtipos se solapan o son disjuntos, y si la unión de los subtipos recubre o no al supertipo, se pueden distinguir cuatro clases de generalización.

Si un mismo ejemplar del supertipo puede pertenecer a más de un subtipo habrá solapamiento, y si sólo puede pertenecer a uno de los subtipos existirá exclusividad; por otro lado, si todo ejemplar del supertipo tiene que pertenecer a algún subtipo tendremos totalidad, y si, por el contrario, no tiene obligatoriamente que pertenecer a algún subtipo habrá parcialidad.

La combinación de estas posibilidades da lugar a cuatro tipos de jerarquía, donde se representa por un arco el hecho de que los subtipos sean disjuntos y con un círculo la presencia de una jerarquía total, como puede observarse en la figura 2.23.

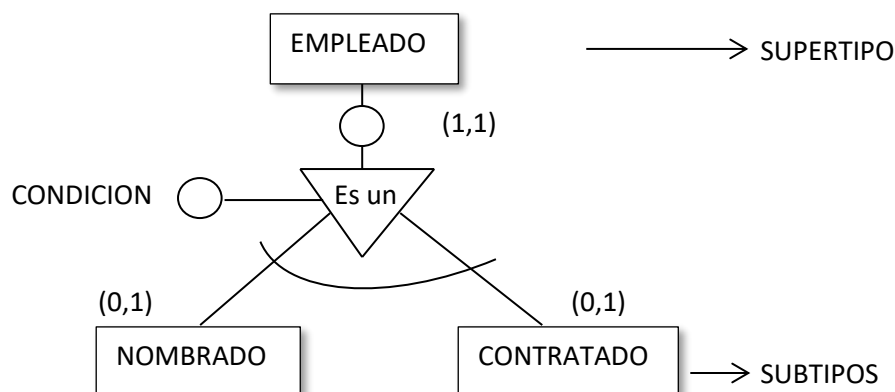


Figura 2.23. Jerarquía total sin solapamiento

2.8. Cómo desarrollar un Esquema Entidad-Interrelación

Una aproximación utilizada habitualmente en la construcción de esquemas E/R es identificar primero las entidades, luego las interrelaciones y por último los atributos.

A veces también se suele recurrir a otro tipo de herramientas que nos ayudan a detectar información que no aparece explícitamente representadas en el enunciado y que resultan de gran utilidad a los diseñadores inexpertos. A continuación se sugieren una serie de pasos que se podrían seguir para este propósito:

- Estudiar el enunciado que describe el Universo del Discurso y elaborar dos listas; una con los candidatos a ser entidades y otras con las posibles interrelaciones junto con su tipo de correspondencia (1:1, 1:N, N:M).
- Construir una Matriz de Entidades en la que las filas y las columnas son nombres de entidades y cada celda puede contener o no nombre de interrelaciones.

	E1	E2	E3	...	EN
E1	I1	I2	--	--	I3
E2	×	I4	I5	--	--
E3	×	×	I6	--	--
...	×	×	×	--	IN
EN	×	×	×	×	×

E1, E2, ... EN son entidades, I1, I2,... IN son interrelaciones. Como la matriz es simétrica, las celdas que aparecen con una cruz se corresponden con interrelaciones que ya han sido especificadas. El símbolo -- en una celda indica que no existe interrelación. Además se debe indicar el tipo de correspondencia de cada interrelación.

- Con la matriz de entidades se construye un primer esquema E/R, con sus entidades, interrelaciones, atributos y sus tipos de correspondencia, además se le añaden las cardinalidades mínimas y máximas.
- En el último paso se refina el esquema estudiando las posibles redundancias siempre y cuando existan ciclos con interrelaciones semánticamente equivalentes

Ejercicio resuelto

Supongamos el siguiente universo del discurso sobre municipios, viviendas y personas. Cada persona sólo puede habitar en una vivienda y estar empadronada en un municipio, pero puede ser propietaria de varias viviendas. Nos interesa también conocer las personas que dependen del Cabeza de Familia (C.F.). Se indicarán los supuestos semánticos que se consideren oportunos, para justificar las decisiones de diseño.

1er paso: Elaborar la lista de candidatos a ser entidades en interrelaciones

Entidades: MUNICIPIO, VIVIENDA, PERSONA

Interrelaciones: **Habita** entre PERSONA y VIVIENDA

Empadronada entre PERSONA Y MUNICIPIO

Propiedad entre PERSONA y VIVIENDA

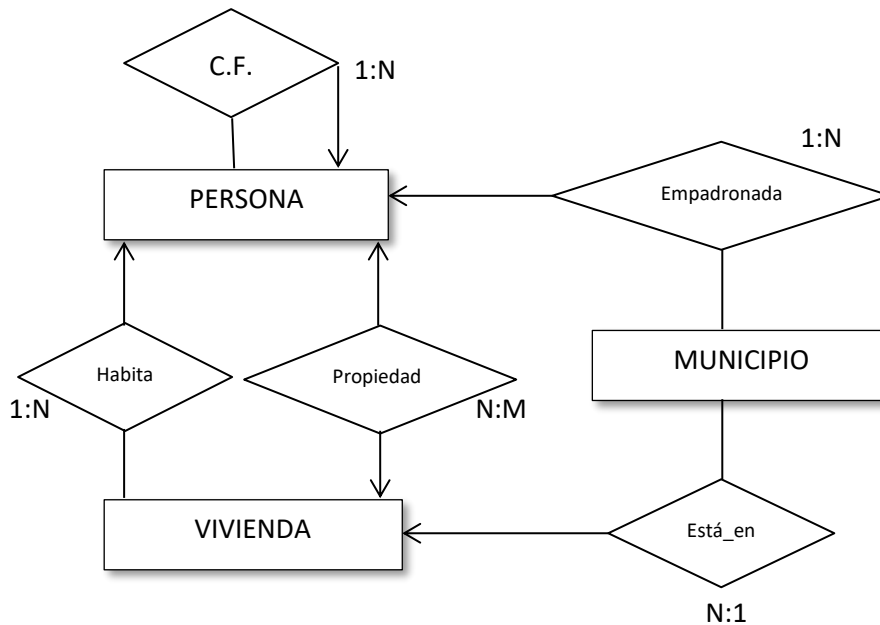
Falta representar el CABA DE FAMILIA, que no sabemos si es una entidad (ya que también es una persona), o un atributo o también podría ser una interrelación. Más adelante veremos cómo clasificarlo.

2do paso: Construir una matriz Entidades/Entidades para representar todas las interrelaciones junto con su tipo de correspondencia. Para ello, iremos analizando los supuestos semánticos explícitos que se encuentran en el enunciado así como aquellos implícitos o que son de sentido común.

- 1) Supuestos dados en el enunciado (Explícitos):
 - Cada PERSONA sólo puede Habitar en una Vivienda
Interrelación **Habita** (1:?) entre PERSONA y VIVIENDA
 - Cada PERSONA puede ser Propietaria de más de una vivienda
Interrelación **Propiedad** (? :N) entre PERSONA y VIVIENDA
 - Las PERSONAS dependen del cabeza de familia.
Interrelación **C.F.** (? :?) entre PERSONA y PERSONA (recursiva)
 - Una PERSONA está empadronada en único MUNICIPIO
Interrelación **Empadronada** (1:N) entre PERSONA y MUNICIPIO
 - 2) Supuestos no dados en el enunciado (Implícitos y de sentido común)
 - En una VIVIENDA pueden Habitar muchas Personas (supuesto lógico del mundo real).
Interrelación **Habita** (1:N) entre PERSONA y VIVIENDA
 - Una VIVIENDA puede ser Propiedad de muchas Personas (supuesto legal)
Interrelación **Propiedad** (M:N) entre PERSONA y VIVIENDA
 - Una PERSONA sólo puede tener un cabeza de familia y un cabeza de familia puede serlo de varias PERSONAS
Interrelación **C.F.** (1:N) entre PERSONA Y PERSONA
 - Un MUNICIPIO puede tener muchas VIVIENDAS y una VIVIENDA pertenece a un solo MUNICIPIO.
Interrelación **Está_En** (N:1) entre MUNICIPIO y VIVIENDA.
- Se obtiene la siguiente matriz:

	PERSONA	MUNICIPIO	VIVIENDA
PERSONA	C.F. (1:N)	Empadronada (1:N)	Habita (1:N) Propiedad (N:M)
MUNICIPIO	×	--	Está_En (N:1)
VIVIENDA	×	×	--

3er paso: Obtener una versión preliminar del esquema E/R.



4to paso: Análisis de las cardinalidades mínimas. Hasta ahora hemos determinado las cardinalidades máximas, que se corresponden con los tipos de correspondencia. Ahora determinaremos las cardinalidades mínimas.

- Interrelación **C.F.**: Una PERSONA tiene obligatoriamente como mínimo una PERSONA que es Cabeza de Familia y una PERSONA que es Cabeza de Familia puede que no tenga ninguna persona a su cargo.
- Interrelación **Habita**: Una PERSONA habita como mínimo en una VIVIENDA y en una VIVIENDA puede que no habite ninguna PERSONA.
- Interrelación **Propiedad**: Una PERSONA puede que no sea propietaria de ninguna VIVIENDA y una VIVIENDA puede que no sea propiedad de ninguna PERSONA (una vivienda podría ser propiedad de una empresa, por ejemplo).
- Interrelación **Empadronada**: Una PERSONA está empadronada como mínimo en un MUNICIPIO (y como máximo también) y en un MUNICIPIO como mínimo está empadronada una PERSONA.
- Interrelación **Está_en**: Una VIVIENDA está en único MUNICIPIO y en un MUNICIPIO hay, como mínimo, una VIVIENDA.

5to paso: Análisis de redundancias. Como existen dos ciclos en el esquema E/R hay que analizar si existe alguna interrelación redundante.

El primer ciclo lo constituyen las interrelaciones **Propiedad**, **Está_En** y **Empadronada**. La primera condición es saber si las interrelaciones son semánticamente equivalentes. En este caso la interrelación **Propiedad** no es semánticamente equivalente a **Está_en** y **Empadronada**, puesto que el poseer o no una vivienda no influye en si la persona reside en el municipio en el que se encuentra la vivienda.

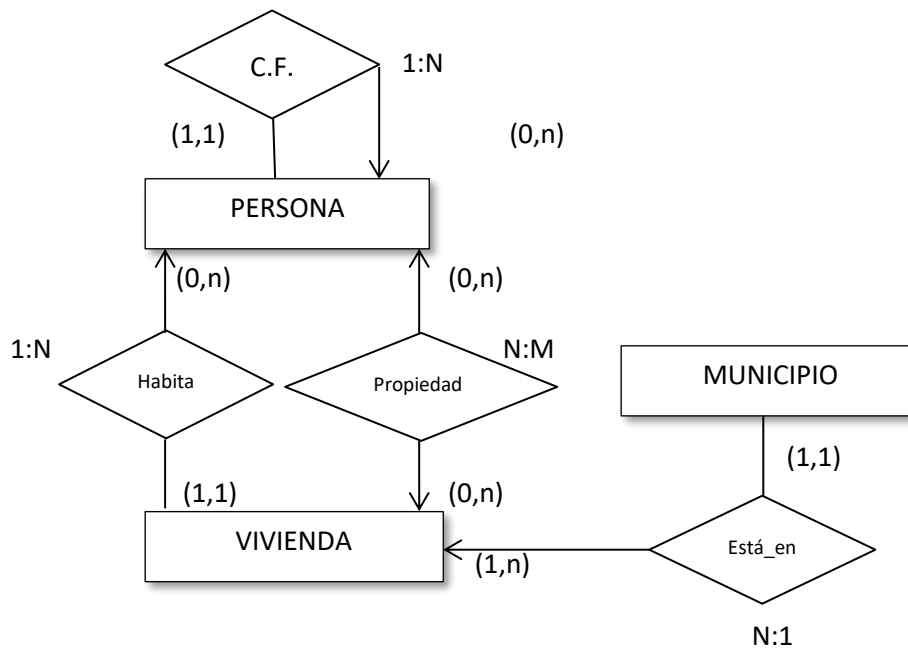
El segundo ciclo lo constituyen las interrelaciones **Habita**, **Está_en** y **Empadronada**. Como las tres interrelaciones están semánticamente relacionadas debemos analizar si existe redundancia.

Interrelación **Habita**: Para eliminarla debe ser posible obtener su semántica a partir de las otras dos interrelaciones. Si queremos obtener las personas que habitan en una determinada vivienda, a partir de **Está_en** se obtiene el municipio en el que se encuentra la vivienda y con la interrelación **Empadronada** se obtiene las personas que habitan ese municipio, pero no sabemos las personas que habitan en todas las viviendas del municipio. Por ello, la interrelación **Habita** no se puede eliminar.

Interrelación **Está_en**: Para eliminar esta interrelación debe ser posible obtener su semántica a partir de las interrelaciones **Habita** y **Empadronada**. Para conocer las viviendas que se encuentran en un determinado municipio, a partir de **Empadronada** obtenemos todas las personas empadronadas en ese municipio y mediante la interrelación **Habita** obtenemos las viviendas en las que habitan esas personas (pues una persona obligatoriamente debe habitar en una vivienda); de esta forma, sabremos las viviendas de ese municipio. En el otro sentido de la interrelación **Está_en**, para conocer en qué municipio está una determinada vivienda, a partir de **Habita** obtenemos las personas que habitan en ella; sin embargo, puede ocurrir que en una determinada vivienda no habite nadie, por lo que no podemos alcanzar la interrelación **Empadronada** entre persona y municipio. Así, la interrelación **Está_en** no es redundante.

Interrelación **Empadronada**: Si eliminamos la interrelación **Empadronada**, debería ser posible obtener su semántica a partir de **Habita** y **Está_en**. Para conocer el municipio en que está empadronada una persona, mediante **Habita** obtenemos la vivienda en la que habita esa persona y con la interrelación **Está_en** obtenemos el municipio en que se encuentra la vivienda. En el otro sentido la interrelación **Empadronada**, debe ser posible conocer las personas empadronadas en un determinado municipio; mediante la interrelación **Está_en** conocemos las viviendas de ese municipio y a partir de **Habita** sabemos todas las personas que viven en esa vivienda, conociendo así todas las personas empadronadas en el municipio. Consecuentemente, la interrelación **Empadronada** se puede eliminar del esquema E/R sin perder semántica.

El esquema E/R definitivo, que incluye las cardinalidades es el siguiente:



Ejercicios propuestos

- 1) El departamento de formación de una empresa desea construir una base de datos para planificar y gestionar la formación de sus empleados.
La empresa organiza cursos internos de formación de los que se desea conocer el código del curso, el nombre, una descripción, el número de horas de duración y el costo del curso.
Un curso puede tener como prerequisite haber realizado otro(s) previamente, y a su vez la realización de un curso puede ser prerequisite de otros. Un curso que es prerequisite de otro puede serlo de forma obligatoria o sólo recomendable.
Un mismo curso tiene diferentes ediciones, es decir, se imparte en diferentes lugares, fechas y con diferentes horarios (intensivo, de mañana o de tarde), En una misma fecha de inicio sólo puede impartirse una edición de un curso.
Los cursos se imparten por personal de la propia empresa.
De los empleados se dese almacenar su código de empleado, nombre y apellidos, dirección, teléfono, DNI, fecha de nacimiento, nacionalidad, sexo, firma y salario, así como si está o no capacitado para impartir cursos.
Un mismo empleado puede ser docente en una edición de un curso y alumno en otra edición, pero nunca puede ser ambas cosas a la vez (en una misma edición de curso o lo imparte o lo recibe).
- 2) El club de ajedrez Genios Peruanos de la ciudad de Lima, ha sido encargado por la federación Internacional de Ajedrez de la organización de los próximos campeonatos mundiales que se celebrarán en Perú. Por este motivo, desea llevar a una base de datos toda la gestión relativa a participantes, alojamientos y partidas. Teniendo en cuenta que:

En el campeonato participan jugadores y árbitros, de ambos se requiere conocer el número de asociado, nombre dirección, teléfono de contacto y campeonato en los que ha participado (como jugador o como árbitro). De los jugadores se precisa además el nivel de juego en una escala del 1 al 10. Ningún árbitro puede participar como jugador.

Los países envían al campeonato un conjunto de jugadores y árbitros, aunque no todos los países envían participantes. Todo jugador y árbitro es enviado por un único país. Un país puede ser representado por otro país.

Cada país se identifica por un número correlativo según su orden alfabético e interesa conocer además de su nombre, el número de clubes de ajedrez existentes en el mismo.

Cada partida se identifica por un número correlativo (Cod_p), la juegan dos jugadores y la arbitra un árbitro. Interesa registrar las partidas que juega cada jugador y el color (blancas o negras) con el que se juega. Ha de tenerse en cuenta que un árbitro no puede arbitrar a jugadores enviados por el mismo país que le ha enviado a él.

Todo participante participa en al menos una partida.

Tanto jugadores como árbitros se alojan en uno de los hoteles en los que se desarrollan las partidas, se desea conocer en qué hotel y en qué fecha se ha alojado cada uno de los participantes. Los participantes pueden no permanecer en Lima durante todo el campeonato, sino acudir cuando tienen que jugar alguna partida alojándose en el mismo o distinto hotel. De cada hotel, se desea conocer el nombre, la dirección y el número de teléfono.

El campeonato se desarrolla a lo largo de una serie de jornadas (día, mes, año) y cada partida tiene lugar en una de las jornadas aunque no tengan lugar partidas todas las jornadas.

Cada partida se celebra en una de las salas de las que pueden disponer los hoteles, se desea conocer el número de entradas vendidas en la sala de cada partida. De cada sala, se desea conocer la capacidad y medios de que dispone (radio, televisión, multimedia...) para facilitar la retransmisión de los encuentros. Una sala puede disponer de varios medios distintos.

De cada partida se pretende registrar todos los movimientos que la componen, la identificación de movimiento se establece en base a un número de orden dentro de cada partida; para cada movimiento se guardan la jugada (5 posiciones) y un breve comentario realizado por un experto.

- 3) Una empresa dedicada a comercializar cocinas desea aumentar su control sobre aquellos elementos que le afectan. Del resultado del análisis que realizan obtienen la siguiente información:

Hay una serie de fabricantes de muebles de cocina. De cada fabricante se dispone de un nombre, una dirección y una relación de números de teléfono.

Cada uno de ellos fabrica varios muebles de cocina. Un mueble de cocina tiene una determinada línea, un determinado color, unas dimensiones dadas (ancho, alto, largo) y puede tener una de las siguientes categorías excluyentes: mueble alto, mueble bajo, panel y encimera. De los muebles bajos interesa saber la

altura sobre el suelo y de las encimeras interesa saber su tipo (mármol o aglomerado).

Cada fabricante puede trabajar con varios distribuidores y cada distribuidor trabaja al menos con un fabricante. De un distribuidor se dispone del nombre, dirección y una relación de números de teléfono.

Una cocina la componen una serie de muebles de cocina de distinto tipo, cada mueble de cocina sólo podrá formar parte de una única cocina. De una cocina nos interesa saber el número de muebles que la componen así como cuántos de ellos ha y de cada tipo.

Cada cocina la puede vender un único distribuidor en una determinada fecha de venta, aunque cada distribuidor puede vender varias cocinas. Un

distribuidor puede ceder una cocina a otro, para que éste pueda venderla.

Cada cocina la debe armar al menos un armador, y el mismo armador puede armar varias cocinas. De un cliente nos interesa su DNI, su nombre, dirección y un único número de teléfono.

- 4) Se desea diseñar una base de datos sobre la información de las reservas de una empresa dedicada al alquiler de automóviles teniendo en cuenta que:

Un determinado cliente puede tener en un momento dado hechas varias reservas. De cada cliente se desea almacenar su DNI, dirección y teléfono.

Además dos clientes se diferencian por un código único.

Cada cliente puede ser avalado por otro cliente de la empresa.

Una reserva la realiza un único cliente pero involucrar a varios autos.

Es importante registrar la fecha de inicio y final de la reserva, el precio del alquiler de cada uno de los autos, los galones de gasolina en el depósito en el momento de realizar la reserva, el precio total de la reserva y un indicador de si el auto o los autos han sido entregados.

No se mantienen los datos de reservas anteriores.

Todo auto tiene siempre asignado un determinado garaje que no puede cambiar. De cada auto se requiere la matrícula, el modelo, el color y la marca.

Cada reserva se realiza en una determinada agencia.

CAPÍTULO 3. MODELO DE DATOS RELACIONAL

3.1. Presentación del Modelo

Este modelo fue propuesto por Codd en el año 1970. Está basado en la teoría matemática de las relaciones, los datos se estructuran lógicamente en forma de relaciones – tablas -. La características más importantes que incorpora este modelo son:

- Sencillez y uniformidad
- Sólida fundamentación teórica
- Interdependencia de la interfaz de usuario.

3.2. Elementos permitidos

3.2.1. Dominios, Relaciones y Atributos

El dominio (D) es un conjunto nominado, finito y homogéneo de valores atómicos. Se especifica mediante un nombre y un formato, el cual puede definirse por extensión (dando sus posibles valores) o por intensión (especificando el tipo de dato).

Un atributo (A) es la interpretación de un determinado dominio en una relación. Si D es el dominio de A, se denota:

Matemáticamente, una relación definida sobre un conjunto de dominios $D_1 \dots D_n$ es un subconjunto del producto cartesiano de los n dominios (n es el grado de la relación).

Un esquema de relación se compone de un nombre de relación R , de un conjunto de n atributos y de un conjunto de n dominios, donde cada atributo será definido sobre un dominio.

$$R (A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$$

Una relación $r(R)$ es un conjunto de m elementos denominados tuplas $\{ t_j \}$:

$$r(R) = t_j \{ \langle A_1 : v_{1j} \rangle, \dots, \langle A_i : v_{ij} \rangle, \dots, \langle A_n : v_{nj} \rangle : v_{ij} \in D_i \}$$

Donde A_i es el nombre de un atributo y v_{ij} es un valor del correspondiente dominio D_i sobre el que está definido el atributo.

Una relación se representa utilizando una tabla, donde:

- Las columnas de la tabla son los atributos que expresan las propiedades de la relación. El número de atributos se llama **grado** de la relación.
- Cada fila de la tabla se llama tupla y es un elemento del conjunto que es la relación. El número de tuplas se llama cardinalidad de la relación. La cardinalidad varía en el transcurso del tiempo.

3.3. Intensión y Extensión de una Relación

De acuerdo con la definición de relación y esquema de relación dadas anteriormente se pueden distinguir dos conceptos ligados a la noción de la relación:

- *Intensión de una relación*: Parte definitoria y estática (invariante en el tiempo) de la relación, es lo que se llama esquema de relación.
- *Extensión*: Conjunto de tuplas que, en un instante determinado, satisfacen el esquema de relación y se encuentran almacenadas en la base de datos, es lo que se suele llamar, simplemente, relación. La extensión varía en el transcurso del tiempo.

La intención de una base de datos relacional, llamada **esquema relacional**, está compuesta por una colección de esquemas de relación que describen un determinado universo del discurso, la extensión del esquema relacional, constituido por una colección de relaciones, es la base de datos relacional.

3.4. Elementos no permitidos: Restricciones

Los elementos no permitidos en la estructura del modelo relacional son de dos tipos: restricciones inherentes y restricciones semánticas.

3.4.1. Restricciones inherentes

Son las restricciones derivadas de la misma estructura del modelo, que no tienen que ser definidas por el usuario e imponen limitaciones a la hora de modelar nuestro mundo real.

La estructura del modelo relacional no permite distinguir ente objetos y asociaciones.

Otra restricción es propia de la definición matemática de la relación, que al definirse como un conjunto, no permite la existencia de tuplas duplicadas y de ahí la obligatoriedad de una clave primaria o identificador (conjunto mínimo de atributos que identifican de forma unívoca la tuplas de una relación).

Otra restricción inherente impuesta por la definición de la relación matemática prohíbe que en el cruce una fila y columna haya más de un valor, es decir no se admiten grupos repetitivos.

3.4.2. Restricciones semánticas

Las restricciones semánticas son facilidades proporcionadas por el modelo a fin de poder recoger en el esquema la semántica del universo del discurso que se está modelando. Son restricciones que tiene que definir el diseñador a fin de que el esquema sea un reflejo lo más fiel posible del mundo real.

En el modelo relacional una restricción de integridad es una regla ECA (Evento-Condición-Acción), donde el evento es una operación de actualización (inserción, borrado o modificación), la condición puede definirse como un

predicado sobre un conjunto de atributos, tuplas o de dominios, que debe ser verificado por los correspondientes elementos para que constituyan una extensión válida del esquema, y la acción puede ser de rechazo de la operación o cualquier otra, determinada por el modelo. La acción se lleva a cabo si una operación de actualización intenta violar la condición.

3.4.2.1. Clasificación según los elementos a los que afecta la condición

- a) Restricciones intrarrelación sobre atributos: Se define a nivel de esquema de relación, como el par $S1 = \langle A, P \rangle$, donde:
 A , es el conjunto de atributos
 P , es un predicado definido sobre A .
 Entre estas restricciones están las de unicidad de uno o varios atributos, cuyos valores no pueden repetirse en las distintas tuplas de una relación. En SQL: UNIQUE $\langle A \rangle$. La acción cuando no se cumple esta condición es de rechazo de la operación.
 Otra restricción de este tipo es la obligatoriedad o prohibición de nulos en un atributo. En SQL: $\langle \text{nombre del atributo} \rangle$ NOT NULL. La acción también en este caso es de rechazo.
 También son restricciones de este tipo la definición de clave primaria, las restricciones de verificación (CHECK).
- b) Restricción intrarrelación sobre tuplas. Se define a nivel de esquema de relación como el par $S2 = \langle T, P \rangle$, donde:
 T es el conjunto de tuplas involucrados en la relación.
 P es un predicado definido sobre T .
 Por ejemplo, la comprobación de que la cardinalidad de una relación no sobrepase un determinado valor, exigiría una condición de este tipo.
- c) Restricción interrelación. Se define en el esquema relacional, como el par $S3 = \langle A, P \rangle$, donde:
 A es el conjunto de atributos pertenecientes a más de una relación denominado contexto de la restricción,
 P es un predicado definido sobre A .
 Restricciones de este tipo son las claves ajenas y las aserciones que se definen sobre más de una tupla.
- d) Restricción sobre dominios: se define en el esquema relacional como el par $S4 = \langle D, P \rangle$, donde:
 D es el conjunto de dominios involucrados en la restricción
 P es un predicado definido sobre D
 Un ejemplo de esta restricción es la verificación sobre los valores de un dominio

3.4.2.2. Restricciones de condición y acción específicas

Entre estas restricciones se encuentran las de clave candidata y ajena.

Clave candidata

Una clave candidata es un conjunto no vacío de atributos (descriptor) que identifican unívoca y mínimamente cada tupla de una relación.

Una relación puede tener más de una clave candidata, entre ellas cabe distinguir:

Clave primaria: Es la clave candidata elegida para identificar las tuplas de la relación.

Claves alternativas: Son aquellas claves candidatas que no han sido elegidas como claves primarias de la relación.

El comando en SQL para crear una clave primaria es: PRIMARY KEY <nombre de los atributos que forman la clave primaria>

Para definir los atributos como claves alternativas se usa el comando UNIQUE.

Clave ajena

Una clave ajena de una relación R2 es un conjunto no vacío de atributos cuyos valores han de coincidir con los valores de la clave primaria de una relación R1 (R1 y R2 pueden ser la misma relación). Se dice que R2 es la relación que referencia, mientras que R1 es la relación referenciada. En la figura 3.1 se muestra un ejemplo, donde los valores del atributo codFacultad de la relación **Alumno** deben coincidir con los de la clave primaria de la relación **Facultad**.

Integridad referencial

Dentro de las restricciones propias del modelo relacional se encuentra la de integridad referencial, que es la condición y acción específica que afirma: “Si una relación R2 tiene un descriptor CA que es clave ajena que referencia a la clave primaria CP de la relación R1, todo valor CA debe coincidir con un valor de CP, o ser nulo. Expresada como predicado sería:

$$R2.CA = R1.CP$$

Alumno

CodAlumno	Nombre	CodFacultad
1025684189	Pedro Feria Ruesta	01
1065842569	Ana Calle Estrada	02
1035475524	Mario Torres Eche	01

Facultad

CodFacultad	Nombre
01	Ingeniería Industrial
02	Economía
03	Ciencias

Figura 3.1. Ejemplo de clave ajena.

La acción es de tipo específico (aunque no es de rechazo en todos los casos). Si se intenta insertar una tupla en la tabla que referencia R2, que no cumpla la condición, la acción es de rechazo. Si la condición falla debido a una operación de borrado de tuplas o de modificación de la clave primaria en la tabla referenciada R1. Existe en SQL la posibilidad de elegir entre cuatro opciones, tanto para la operación de borrado como para la modificación:

NO ACTION (rechazar la operación)

CASCADE (propagar la modificación o borrar las tuplas de la tabla que referencia)

SET NULL (poner valor nulo en la CA de la tabla que referencia)

SET DEFAULT (poner valor por defecto en la CA la tabla que referencia)

3.4.2.3. Restricciones de Condición General (Predicado Libre) y de Acción Específica (Rechazo)

Entre las restricciones de este tipo se encuentran las de verificación ("CHECK") y las aserciones ("ASSERTION").

3.4.2.4. Restricciones de Condición General y Acción General (Disparadores)

Los disparadores (triggers) son procedimientos que el usuario define para ejecutar una acción que ha de aplicarse cuando se cumple una condición.

Este tipo de restricciones es muy importante en el diseño de base de datos a fin de perder semántica del esquema conceptual que no es posible recoger mediante las anteriores restricciones.

CAPÍTULO 4. DEPENDENCIAS FUNCIONALES

4.1. Dependencias entre los datos

Las dependencias son propiedades inherentes al contenido semántico de los datos; forman parte de las restricciones de usuario del modelo relacional se han de cumplir para cualquier extensión de un esquema. Por simplificación se considera que el esquema relacional está compuesto por un único esquema, el cual es un par de la forma: $R(A, DEP)$, donde:

A es el conjunto de atributos de la relación

DEP es el conjunto de dependencias existentes entre los atributos.

Existen distintos tipos de dependencias: funcionales, multivaluadas, jerárquicas y de combinación. Cada tipo de dependencia constituye un caso particular del grupo que le sigue; así las dependencias funcionales son un caso particular de las dependencias multivaluadas, éstas de las jerárquicas y estas últimas de las dependencias de combinación.

El grupo más restrictivo y a la vez más numeroso de asociaciones entre los datos es el de las dependencias funcionales. Sobre este conjunto de dependencias se apoyan las tres primeras formas normales y la forma normal de Boyce_Codd, que veremos más adelante.

4.2. Concepto de Dependencia Funcional

Sea el esquema de relación $R(A, DF)$ y sean X e Y dos **descriptores**. Se dice que existe una **dependencia funcional entre X e Y** de forma que X determina a Y , si, y sólo si, se cumple que para cualesquiera dos tuplas de R , u y v tales que $u[X] = v[X]$, entonces necesariamente $u[Y] = v[Y]$ (la notación $u[X]$ o $v[X]$ indica la proyección de la tupla u o de la tupla v , respectivamente, sobre el descriptor X , cuyo resultado será un valor de X); análogamente $u[Y]$ o $v[Y]$ indica la proyección de u o v , respectivamente sobre el descriptor Y .

Un **determinante** o **implicante** es un conjunto de atributos del que depende funcionalmente otro conjunto de atributos al que llamamos **implicado**.

Por ejemplo, podemos decir que el DNI de una persona determina su nombre.

$$\text{DNI} \rightarrow \text{Nombre}$$

Dos descriptores X e Y se dicen que son equivalentes si

$$X \rightarrow Y \wedge Y \rightarrow X$$

Lo que también se representa:

$$X \leftrightarrow Y$$

Por ejemplo, en la relación *Alumno* los atributos *codAlumno* y *DNI* son equivalentes, ya que el código de alumno determina su DNI y a la vez el DNI determina el código del alumno, y se representa:

$\text{codAlumno} \leftrightarrow \text{DNI}$

Cabe aclarar que el hecho de que el código del alumno determina el DNI no significa que dado el código de alumno podamos deducir a partir de él cuál es su DNI, a no ser que tengamos la extensión *r* del esquema de relación que contiene la correspondiente dependencia funcional.

Un ejemplo aclarará esto. Supongamos que se tiene una relación *AlumnoCurso*, del cual mostramos algunas columnas y sus correspondientes valores de las tuplas.

AlumnoCurso

	CodAlumno	DNI	Curso
1	1050677823	02657865	SI5467
2	1045673839	¿??????	SG4356
3	1035763457	02645367	SI3265
4	1050677823	¿??????	SG7890

En la tupla 2 tenemos un DNI desconocido y no lo podemos deducir tan sólo conociendo cuál es su código de alumno; sin embargo, el valor del DNI de la tupla 4 sí se puede deducir dado que en la tupla 1 tenemos el valor del DNI que corresponde a ese código de alumno, y dada la dependencia funcional que existe entre ambos podemos afirmar que el DNI es 02657865.

4.2.1. Dependencia funcional plena o completa

Se dice que *Y* tiene dependencia funcional plena o completa de *X*, si depende funcionalmente de *X*, pero no depende de ningún subconjunto de éste.

Se representa por $X \Rightarrow Y$

Por tanto,

$X \Rightarrow Y$ si $\nexists X' \subset X \mid X' \rightarrow Y$

Ejemplo: Supongamos la siguiente relación:

Notas (CodAlumno, Semestre, CodCurso, Nota)

En la cual se registran las notas de los alumnos por semestre y curso, y se da la siguiente dependencia funcional:

$$\text{CodAlumno, Semestre, CodCurso} \rightarrow \text{Nota}$$

Sin embargo:

$$\nexists X' (\text{CodAlumno, Semestre, CodCurso}) \mid X' \rightarrow \text{Nota}$$

Es decir Nota no depende funcionalmente de ningún subconjunto de CodAlumno, Semestre, CodCurso, es decir tiene dependencia funcional plena con respecto a estos atributos, esto es:

$$\text{CodAlumno, Semestre, CodCurso} \Rightarrow \text{Nota}$$

4.2.2. Dependencia Funcional trivial

Una dependencia funcional $X \rightarrow Y$ se dice que es trivial si Y es un subconjunto de X ($Y \subseteq X$). Por ejemplo, serán triviales las siguientes dependencias:

$$\text{CodAlumno} \rightarrow \text{CodAlumno}$$

$$\text{CodCurso, Semestre} \rightarrow \text{CodCurso}$$

4.2.3. Dependencia funcional elemental

Se dice que una dependencia funcional $X \rightarrow Y$ es elemental si Y es un atributo único no incluido en X, y no existe X' incluido en X tal que $X' \rightarrow Y$, es decir, la dependencia funcional elemental es una dependencia funcional completa, no trivial y en la que el implicado es un atributo único:

$$X \rightarrow Y \text{ es elemental si } (\nexists Y' \subset Y) \wedge (Y \subseteq X) \wedge (\nexists X' \subset X \mid X' \rightarrow Y)$$

Este tipo de dependencias son las que se utilizarán para la teoría de la normalización.

4.2.4. Dependencia funcional transitiva

Dado el esquema de relación:

$$R(X, Y, Z)$$

En el que existen las siguientes dependencias funcionales:

$$X \rightarrow Y$$

$$Y \rightarrow Z$$

$$Y \not\rightarrow X$$

Se dice que Z tiene **dependencia transitiva** respecto a X a través de Y, lo que se representa por:

$$X \multimap Z$$

Si, además $Z \not\rightarrow Y$ se dice que la dependencia transitiva es **estricta**.

Ejemplo:

Consideremos una relación CursoPrograma, la cual contiene los cursos de programas de Patpro:

CursoPrograma (CodCurso, CodPrograma, CodDepartamento)

En donde tenemos que para cada curso su código, el programa que lo incluye y el departamento del que depende el programa (suponiendo que un curso se imparte en un único programa y que un programa lo prepara un único departamento y además que en un programa se pueden dictar varios cursos), se tendrán las siguientes dependencias funcionales:

$$\text{CodCurso} \rightarrow \text{CodPrograma}$$
$$\text{CodPrograma} \rightarrow \text{CodDepartamento}$$

Además:

$\text{CodPrograma} \not\rightarrow \text{CodCurso}$ (ya que en un programa se dictan varios cursos)

Entonces:

$$\text{CodCurso} \rightarrow \text{CodDepartamento}$$

Por tanto, la dependencia funcional entre CodCurso y CodDepartamento es una dependencia transitiva a través de CodPrograma, que se representa por:

$$\text{CodCurso} \multimap \rightarrow \text{CodDepartamento}$$

Además, como $\text{CodDepartamento} \not\rightarrow \text{CodPrograma}$, se da una dependencia transitiva estricta.

CAPÍTULO 5. TEORÍA DE LA NORMALIZACIÓN

La Teoría de la Normalización permite una formalización en el diseño lógico de bases de datos relacionales, lo que permite disponer de instrumentos algorítmicos de ayuda al diseño y, de esta forma, poder desarrollar programas que automaticen el diseño en el modelo relacional.

En este capítulo se estudiará la teoría de la normalización que permite afrontar el problema de diseño de base de datos relacionales de una manera rigurosa y objetiva, estudiando la forma de llevar a cabo la normalización de un esquema relacional basándonos en el concepto de dependencias funcionales visto en el capítulo anterior.

5.1. Necesidad de un método formal de diseño relacional

Al estudiar la estructura del modelo relacional veíamos que la información de nuestra base de datos se puede representar por medio de un conjunto de objetos (relaciones y dominios) y de un conjunto de reglas de integridad.

Estas relaciones que resultan de la observación del mundo real, pueden presentar algunos problemas, derivados de fallos en la percepción del Universo del Discurso, entre estos problemas cabe destacar los siguientes:

- Incapacidad para almacenar ciertos hechos.
- Redundancias y, por tanto, posibilidad de inconsistencias.
- Ambigüedades
- Pérdida de información.
- Pérdida de dependencias funcionales
- Existencia de valores nulos.
- Aparición, en la base de datos, de estados que no son válidos en el mundo real (anomalías de inserción, borrado y modificación).

Veamos un ejemplo:

Supongamos que tenemos la relación PLANILLA de la figura 5.1.

Codigo	Nombre	Direccion	Concepto	Importe
00525	PEDRO MENDOZA FERIA	AH SANTA ROSA R-15	BASICO	1500
00525	PEDRO MENDOZA FERIA	AH SANTA ROSA R-15	FAMILIAR	75
00486	ANA PALACIOS CALLE	SANTA ISABEL L-10	BASICO	1200
00486	ANA PALACIOS CALLE	SANTA ISABEL L-10	FAMILIAR	84
00126	MIGUEL LOPEZ CASTRO	SANTA ANA S-18	BASICO	1500
00126	MIGUEL LOPEZ CASTRO	SANTA ANA S-18	FAMILIAR	85
00525	PEDRO MENDOZA FERIA	AH SANTA ROSA R-15	ESPECIAL	350

Figura 5.1. Ejemplo de diseño inadecuado

En este ejemplo inadecuado de diseño vemos que se presentan varios de los problemas enumerados anteriormente:

- Gran cantidad de redundancias; ya que el nombre y dirección de los trabajadores se repite por cada concepto de planilla que se paga.
- Anomalías de modificación; ya que podemos, por ejemplo, cambiar la dirección de un trabajador en una tupla y por error no modificarla en el resto de las que corresponden al mismo trabajador, lo que da lugar a inconsistencias.
- Anomalías de inserción; ya que si quisiera incluir información sobre algún trabajador que no está en planilla no sería posible ya que los trabajadores tienen algún concepto de la planilla.
- Anomalías de borrado; ya que si quisiéramos dar de baja un concepto, también se perderían datos sobre los trabajadores que tienen pagos por ese concepto.

Si se hubiera llevado a cabo un diseño riguroso, no se habría presentado una relación de este tipo.

5.2. Teoría Formal de la Normalización

Dado un conjunto A de atributos y el conjunto D de dependencias existentes entre ellos, que puede considerarse que constituyen un esquema de la relación $R(A, D)$ (esquema origen), se trata de transformar, por medio de sucesivas proyecciones este esquema de partida en un conjunto de n esquemas de relación (esquemas resultantes): $\{R_i(A_i, D_i)\}_{i=1}^n$ tales que cumplan unas determinadas condiciones.

Es evidente que una base de datos no puede estar constituida por una única relación con todos los atributos (la relación universal), ya que ello daría lugar a una enorme cantidad de redundancias, provocando las anomalías de actualización a las que nos hemos referido. Por ello es necesario reemplazar el esquema que contiene todos los atributos y todas las dependencias por un conjunto R_i de esquemas equivalentes que cumplan unas ciertas propiedades, a fin de conseguir la mejor representación de nuestro universo del discurso. Este conjunto de esquemas debe ser **equivalente y mejor** que R , para ello es necesario que se cumplan tres propiedades:

- a) Conservación de la información
- b) Conservación de las dependencias
- c) Mínima redundancias de los datos (normalización de las relaciones)

Si se cumplen las dos primeras propiedades, se dice entonces que R_i es equivalente a R .

Si las relaciones del esquema resultante R_i está en formas normales más avanzadas que el esquema origen R , se dice en este sentido de nivel de normalización que R_i es mejor que R .

A continuación se expone en qué consiste cada una de estas tres propiedades:

5.2.1. Conservación de la información

La finalidad de esta propiedad es conseguir que el proceso de normalización se lleve a cabo sin pérdida de la información existente en la base de datos.

Para que se cumpla esta propiedad son precisas dos condiciones:

a) Conservación de los atributos

El conjunto de atributos de los esquemas resultantes ha de ser igual al conjunto de atributos del esquema origen, lo que podemos expresar:

$$\bigcup_{i=1}^n A_i = A$$

b) Conservación del contenido (de las tuplas)

Para toda extensión r de R , la combinación de las relaciones resultantes r_i ha de producir la relación origen r .

5.2.2. Conservación de las dependencias

La transformación del esquema origen R en un conjunto de esquema R_i debe llevarse a cabo sin pérdida de estas dependencias o, lo que es lo mismo, el conjunto de dependencias funcionales de partida debe ser equivalente al conjunto de dependencias funcionales de los esquemas resultantes.

5.3. Definición Formal de las tres primeras formas normales.

La tercera propiedad que debe cumplir el conjunto R_i de esquemas resultantes en un proceso de descomposición es que estas relaciones alcancen un nivel de normalización superior al del esquema origen R , a fin de eliminar en lo posible las redundancias y, por tanto, las anomalías de actualización.

Se dice que un esquema de relación está en una determinada forma normal si satisface un cierto conjunto específico de restricciones.

Cuanto más alta sea la forma normal en la que se encuentran los esquemas de relación, menores serán los problemas que aparecen en el mantenimiento de la base de datos.

Codd propuso inicialmente tres formas normales basadas en las dependencias funcionales, primera (1FN), segunda (2FN) y tercera forma normal (3FN).

Debido a que aún persistían los problemas en relaciones en 3FN. Codd, en 1974, introdujo una definición más restrictiva de la tercera forma normal, que se denominó Forma Normal Boyce-Codd (FNBC).

Fagin introdujo la cuarta forma normal y la quinta forma normal, en 1977 y 1979, respectivamente, basadas en otro tipo de dependencias la multivaluadas y la de proyección-combinación.

5.3.1. Primera Forma Normal (1FN)

La primera forma normal es una restricción inherente al modelo relacional, por lo que su cumplimiento es obligatorio y afecta al número de valores que pueden tomar los atributos de una relación.

Hay que tener en cuenta que para que una tabla pueda ser considerada como una tabla no debe admitir grupos repetitivos, es decir debe estar en primera forma normal.

En el siguiente ejemplo se puede apreciar una relación pasada a 1FN.

Persona (No está en 1FN)

DNI	Nombre	Teléfono
02605478	JUAN ROBLES ZAPATA	324578 335689
02745879	ANA FERIA CALLE	615478 645278

Persona (Está en 1FN)

DNI	Nombre	Teléfono
02605478	JUAN ROBLES ZAPATA	324578
02605478	JUAN ROBLES ZAPATA	335689
02745879	ANA FERIA CALLE	615478
02745879	ANA FERIA CALLE	645278

Figura 5.2. Relación pasada a 1FN, eliminando grupos repetitivos

Definición:

Se dice que una relación está en 1FN cuando cada atributo sólo toma un valor del dominio subyacente.

5.3.2. Segunda Forma Normal (2FN)

La segunda forma normal está basada en el concepto de dependencia plena y en las interrelaciones existentes entre los atributos principales (que se encuentran en alguna de las claves) y no principales (que no se encuentran en ninguna clave) de una relación.

Definición:

Se dice que una relación está en 2FN si:

- Está en 1FN.
- Cada atributo no principal tiene dependencia funcional completa respecto de cada una de las claves.

La segunda forma normal no se cumple cuando algún atributo no principal depende funcionalmente de algún subconjunto de la clave.

Se puede afirmar que cualquier relación binaria se encuentra siempre en segunda forma normal; así como también está en 2FN cualquier relación en la que todas las claves son simples, es decir contienen un solo atributo.

Asimismo, está en 2FN cualquier relación en la que todos sus atributos son principales, es decir, forman parte de alguna clave.

Siempre es posible transformar un esquema de relación que no esté en 2FN, en esquemas de relación en 2FN, sin que se produzca pérdida de información ni de dependencias.

Sea el esquema de relación ALUMNO_INSCRIBE (AT, DEP), donde:

AT = (CodAlumno, CodCurso, FechaInsc, Escuela)

DEP = (CodAlumno, CodCurso \rightarrow FechaInsc
CodAlumno \rightarrow Escuela)

Que describe las inscripciones de los alumnos, la fecha en que se inscriben y la escuela a la que pertenecen los alumnos.

La clave de la relación es (CodAlumno, CodCurso), se puede observar que la Escuela no es un hecho acerca de la totalidad de la clave, sino acerca de parte de ella (en este caso del atributo CodAlumno). Esta relación no está en 2FN.

Transformamos la relación ALUMNO _INSCRIBE, en las relaciones ALUMNO _INSCRIBE1 y ALUMNO que se encuentran en 2FN:

ALUMNO _INSCRIBE1 (AT1, DEP1), donde

AT1 = (CodAlumno, CodCurso, FechaInsc)

DEP1 = (CodAlumno, CodCurso \rightarrow FechaInsc)

y ALUMNO (AT2, DEP2), donde:

AT2 = (CodAlumno, Escuela)

DEP2 = (CodAlumno \rightarrow Escuela)

5.3.3. Tercera Forma Normal (3FN)

La tercera forma normal (3FN) está basada en el concepto de dependencia transitiva:

Definición:

Un esquema de relación R está en tercera forma normal si, y sólo si:

- Está en 2FN.

- No existe ningún atributo no principal que dependa transitivamente de alguna clave de R.

La tercera forma normal no se cumple cuando existen atributos no principales que dependen funcionalmente de otros atributos no principales.

Se puede afirmar que toda relación binaria se encuentra automáticamente en 3FN, así como toda relación cuyos atributos son todos principales o bien cuando hay un único atributo no principal.

Siempre es posible transformar un esquema de relación que no está en 3FN, sin que se produzca pérdida de información ni de dependencias funcionales.

Sea el esquema de relación ALUMNO (AT, DEP), donde:

AT = (CodAlumno, CodEscuela, NombreEscuela)

DEP = (CodEscuela \rightarrow NombreEscuela)
CodAlumno \rightarrow CodEscuela)

La única clave del esquema de relación es el atributo CodAlumno. El atributo NombreEscuela es un hecho acerca del atributo CodEscuela, atributo que no forma parte de la clave. Por lo tanto, este esquema de relación no está en 3FN (está en 2FN).

Se puede transformar la relación ALUMNO en las relaciones ALUMNO1 y ESCUELA que sí se encuentran en 3FN.

ALUMNO1 (AT1, DEP1), donde:

AT1 = (CodAlumno, CodEscuela)

DEP1 = CodAlumno \rightarrow CodEscuela)

ESCUELA (AT2, DEP2), donde:

AT2 = (CodEscuela, NombreEscuela)

DEP2 = (CodEscuela \rightarrow NombreEscuela)

5.3.4. Forma Normal de Boyce-Codd (FNBC)

Con el paso del tiempo las tres formas normales que acabamos de describir resultaron insuficientes para afrontar ciertos problemas en relaciones que presentaban varias claves candidatas compuestas que se solapaban. Por ello en 1974 Boyce y Codd definieron la forma normal que lleva su nombre (FNBC). Se trata de una redefinición más estricta de la 3FN.

Se dice que una relación se encuentra en FNBC si, y sólo si, todo determinante es una clave candidata.

Supongamos la relación:

ASISTE (CodCurso, NomCurso, CodAlumno, Calificación)

Si consideramos que los nombres de los cursos no pueden repetirse, y que un alumno obtiene una calificación en cada curso al que asiste, tendría las siguientes dependencias funcionales:

$$\text{CodCurso} \leftrightarrow \text{NomCurso}$$
$$\text{CodCurso, CodAlumno} \rightarrow \text{Calificación}$$

Por tanto, habrían dos claves candidatas: (CodCurso, CodAlumno) y (NomCurso, CodAlumno). Esta relación está en 3FN (todos sus atributos, menos uno son principales), sin embargo tiene anomalías de actualización, ya que se repetiría el nombre y el código de los cursos por cada alumno que asistiese a ellos; el problema es debido a que la relación ASISTE no se encuentra en FNBC, ya que tanto CodCurso como NomCurso son determinantes, pero no son claves candidatas de la relación

Se puede afirmar que toda relación binaria está en FNBC.

No siempre es posible transformar un esquema de relación que no está en FNBC en esquema de relación en FNBC sin que se produzca pérdida de dependencias funcionales. Sí se puede asegurar que la transformación se puede producir sin pérdida de información.

Ejercicios Resueltos

1. El Ministerio de Cultura ha decidido guardar información referente a la peregrinación que se realiza a la Cruz de Motupe. La información que se desea almacenar es la siguiente:

Teniendo en cuenta que la peregrinación a Motupe se puede realizar por distintos caminos (camino 1, camino 2, etc), se quiere guardar información acerca de éstos. Esta información se refiere al nombre, el número de kilómetros totales y el tiempo estimado para la realización del camino.

Cada camino se compone de distintas etapas que se identifican por un número correlativo dentro de cada camino, y para cada una de ellas se desea saber el número de kilómetros, el tiempo estimado y las distintas localidades por las que pasa. Además se quiere recoger la localidad de salida y de llegada de la correspondiente etapa.

Se recogerán las distintas localidades por las que pasa cada camino. La información que se recogerá de cada localidad será: nombre de la misma, Comunidad Autónoma a la que pertenece y código postal. Se debe tener en cuenta que pueden existir localidades comunes a distintos caminos.

Se desea guardar información sobre los albergues para peregrinos que existen en algunas localidades que pertenecen al camino. Esta información consta de: nombre del albergue, capacidad y precio.

Por último, se quiere registrar los peregrinos que realizan el camino. Para llevar este control cada uno de ellos lleva un carnet que consta de un número de identificación, el nombre completo del peregrino, su lugar de procedencia y las localidades por las que ha pasado a lo largo del recorrido junto con el día que llegaron a dicha localidad.

Elaborar el esquema relacional indicando las claves primarias y foráneas.

Solución

Camino

idCamino (CP)
nombre
Km
tiempo

Etapas

idEtapa (CP)
idCamino (CF)
km
tiempo

Localidades

idLocalidad (CP)
nombre
comunidad
codPostal

EtapasLocalidades

idEtapa (CP) (CF)
idLocalidad (CP) (CF)
tipo (S: salida, L:llegada, O:Otro)

Albergue

idAlbergue (CP)
idLocalidad (CF)
nombre
capacidad
precio

Peregrino

idCarnet (CP)
nombre
dirección

PeregrinoLocalidad

idCarnet (CP) (CF)
idLocalidad (CP) (CF)
fecha (CP)

2. El Gobierno Regional quiere elaborar una Base de Datos para la administración de la planilla de todos sus trabajadores, para lo cual debe elaborar el esquema relacional de esta BD, identificando las claves primarias y foráneas. Se tienen los siguientes supuestos semánticos:

Cada trabajador se identifica por un código, además se necesitan tener sus datos de nombre, dirección, dni, teléfonos (fijos, celulares, etc), fecha de ingreso, fecha de cese, fecha de nacimiento, sexo, estado civil. También se necesita conocer los datos de sus hijos y cónyuge, de estos se quiere conocer el nombre, dni y fecha de nacimiento.

Los trabajadores pueden ser del tipo empleado, obrero, funcionario y además pueden ser nombrados o contratados. Cada trabajador está asignado a una oficina y además tiene un cargo. A lo largo de su vida laboral pueden ocupar distintos cargos y estar asignados a distintas oficinas, se quiere saber la fecha de inicio y término en cada cargo y oficina.

Las planillas se elaboran mensualmente y pueden ser del tipo normal, vacaciones y gratificaciones. En estas planillas se debe registrar por cada trabajador todos los conceptos de ingresos y egresos y el importe correspondiente. Los conceptos están debidamente identificados con un código, nombre, abreviatura y tipo (ingreso o egreso). En las planillas se debe además identificar la fecha en que fue emitida y por cada trabajador

el cargo y oficina en la que está asignado en el momento en que se emite la planilla.

<u>Trabajador</u>	<u>TipoTelefono</u>	<u>TelefonoTrabajador</u>
codTrab (CP)	idTipoTelf (CP)	CodTrab (CP) (CF)
nombre	nombre	idTipoTelef (CP)
Dirección		numero (CP)
Dni	<u>EstadoCivil</u>	<u>TipoTrabajador</u>
fecIngreso	idEstCiv (CP)	idTipoTrab (CP)
fecCese	nombre	nombre
fecNacim	<u>Oficina</u>	<u>Cargo</u>
idEstCiv (CF)	idOficina (CP)	idCargo (CP)
idTipoTrab (CF)	nombre	nombre
sexo		
<u>TrabajadorOficina</u>	<u>TrabajadorCargo</u>	<u>TipoPlanila</u>
codTrab (CP) (CF)	codTrab (CP) (CF)	idTipoPlan (CP)
idOficina (CP) (CF)	idCargo (CP) (CF)	Nombre
fecInicio (CP)	fecInicio (CP)	
FecFin	fecFin	
<u>TrabajadorFamiliar</u>	<u>Planilla</u>	<u>PlanillaDetalle</u>
codTrab (CP) (CF)	idPlanilla (CP)	idPlanilla (CP) (CF)
Dni (CP)	codTrab (CF)	idConcepto (CP) (CF)
Nombre	Mes	Importe
Tipo	Año	
<u>Conceptos</u>	idtipoPlan (CF)	
idConcepto (CP)	fecEmision	
Nombre	idCargo (CF)	
Abrev	iOficina (CF)	
Tipo		

- Una empresa inmobiliaria alquila puestos comerciales en varios puntos de la ciudad (diferentes ubicaciones). Los puestos se tienen identificados con un código único. Son alquilados a personas o empresas, de las cuales se registra su documento de identidad (DNI para personas y RUC para empresas), el nombre de la persona o razón social de la empresa, la dirección, teléfono. A los inquilinos se les hace un contrato por un periodo determinado, el mismo que no puede ser inferior a 6 meses, pero se incluye en el contrato la posibilidad de rescindirlo (anularlo) por cualquiera de las dos partes (inquilino inmobiliaria) antes de que se cumpla el final del alquiler, se requiere saber la fecha de rescisión y el motivo. Los contratos también se pueden renovar de mutuo acuerdo, para lo cual sólo se requiere conocer la nueva fecha de fin del contrato, y el nuevo importe de alquiler (si este es modificado), tener en cuenta que el contrato se puede renovar varias veces. Cada contrato tiene un número correlativo, el periodo de vigencia, el importe del alquiler, los datos del inquilino. Los importes de alquiler se cobran mensualmente, teniendo como fecha límite de pago el último día no feriado de cada mes, por lo que la empresa requiere tener también un registro de deudas y pagos por

todos sus puestos, en ambas se debe poder identificar el inquilino, el puesto y el importe, tener en cuenta los otros datos que crea necesarios, tales como fecha de vencimiento, fecha de pago, etc. Asimismo, en caso de que el inquilino no cumpla con pagar el importe de alquiler se le cobra interés moratorio por cada día que deje de cancelar, el interés se calcula con una tasa porcentual que se aplica al importe de la deuda. La tasa porcentual de interés puede variar cada mes. Elaborar el esquema relacional que permita modelar este universo del discurso, indicando las claves primarias y foráneas. Además indicar los campos que son obligatorios y los que no lo son.

<u>Puesto</u>		<u>Persona</u>		<u>Lugares</u>	
Código (CP)		idPersona (CP)		idLugar (CP)	
idLugar (CF)		Nombre		nombre	
		Dirección			
		Teléfono			
<u>Contrato</u>		DNI *		<u>ContratoRescinde</u>	
idContrato (CP)		RUC *		idContrato (CP) (CF)	
idPuesto (CF)				Fecha (CP)	
idPersona (CF)				Motivo	
Fecha		<u>ContratoRenueva</u>		<u>Deuda</u>	
fechaIni		idContrato (CP)		idDeuda (CP)	
fechaFin *		Fecha (CP)		idPuesto (CF)	
Importe		fechaFin		idPersona (CF)	
		importe		Mes	
<u>Pago</u>		<u>Tasa</u>		Año	
Recibo (CP)		idTasa (CP)		importe	
idPuesto (CF)		Mes		fecVencim	
idPersona (CF)		Año			
Mes		tasa			
Año					
importe					
fecPago					

Ejercicios Propuestos

1. Se quiere diseñar una base de datos que contenga información sobre las observaciones realizadas a distintas especies de la Selva Peruana. Para ello se han de considerar las siguientes especificaciones:
Cada especie se identifica por su nombre científico. Se desea conocer su nombre vulgar y una breve descripción de las características más importantes que las diferencian del grupo ornitológico al que pertenece. Los grupos ornitólogos clasifican las especies según sus similitudes estructurales, por ejemplo, el grupo de las zancudas se caracteriza por tener patas largas y pico largo. Por lo tanto guardaremos información acerca de las patas, dedos y pico.
Cada especie de ave está siempre asignada a un solo grupo ornitológico.

Existen asociaciones ornitológicas de las cuales se quiere saber su nombre, la dirección y el teléfono.

Cada una de estas asociaciones consta de un grupo de personas, los observadores, que son los que realizan los avistamientos de las distintas especies. De los observadores se requiere su código de observador, el DNI, el nombre completo y la dirección. Además un observador sólo puede estar adscrito a una asociación.

Las zonas de observación tienen un código de zona que es único. También se quiere saber el nombre de la misma, la comunidad autónoma a la que pertenece, la provincia y el tipo (si es una laguna, un río, una zona costera, etc.). En cada zona se encuentra por los menos un observador, y si existe más de uno en la misma zona pueden ser de distintas asociaciones ornitológicas. Cada observador está asignado a una determinada zona.

También se desea guardar información acerca de la fecha en la que el observador divisa una determinada especie, teniendo en cuenta que éstos realizan trabajo de campo cada tres días y que además sólo anotan el primer avistamiento de una determinada especie por día. Esta información es muy importante para controlar las especies de ave que existen en la Selva peruana y, por tanto, tendremos que guardar las observaciones aunque los observadores ya no se encuentren activos. Diseñar el esquema relacional, indicando las claves primarias y ajenas. Marcar con un asterisco los campos que acepten valores nulos.

2. La asociación internacional “Voto Blanco”, tras una larga labor de investigación, ha logrado obtener valiosa información acerca de los numerosos casos de corrupción que se están dando en un país imaginario. Para tenerla toda ordenada y accesible va a preparar una base de datos que contenga todos esos datos.

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

Cada caso es investigado por un juez del que se desea saber su nombre, dirección, fecha de nacimiento y fecha en que comenzó a ejercer. Una vez concluida la investigación del caso emiten un dictamen que se registrará, junto con la fecha de su emisión.

En cada caso hay una serie de ciudadanos implicados, cada uno de ellos con un cargo principal determinado en el momento en que se produjo el caso. De cada uno de ellos se desea conocer su DNI, nombre, dirección y patrimonio.

Estos ciudadanos pueden o no pertenecer a un partido político determinado y, a veces, desempeñan un puesto en él. De cada partido quiere conocerse su nombre, dirección de la sede central y teléfonos, así como el nombre del secretario general de ese partido, tener en cuenta que este secretario puede cambiar en el tiempo.

Cada caso de corrupción es descubierto por un periódico (nombre, dirección y tirada) en una fecha determinada, que se desea saber. Cada

periódico puede tener o no una afinidad con un partido político en un momento determinado (por otro lado un partido político puede tener afinidad con varios periódicos o ninguno).

Diseñe el esquema relacional. Indique en él las claves primarias, claves ajenas. Marque también con un asterisco los campos que aceptan valores nulos.

3. Se desea recoger en una base de datos información acerca del funcionamiento de una comisaría de policía.

Se consideran los siguientes supuestos semánticos:

En la comisaría trabajan una serie de policías (CIP, DNI, nombre, categoría) que pueden desempeñar funciones distintas: administrativos, agentes, etc., las mismas que pueden cambiar en el tiempo.

Cada policía tiene un único jefe, aunque un policía puede ser jefe de varios.

En la comisaría existe un arsenal de armas. Cada arma está identificada por un código único, pertenece a una clase y tiene un nombre determinado.

Cada policía puede utilizar una o varias armas en un momento determinado. Es importante conocer el grado de habilidad (puntuación de 1 a 10) de cada policía con cada una de las armas que utiliza.

Un delincuente (DNI, nombre, alias) es arrestado por uno o varios policías. A cada delincuente que permanezca en la comisaría se le encierra en un calabozo (código y ubicación). En el calabozo pueden estar encerrados varios delincuentes.

Los delincuentes están involucrados en casos (código de caso y juzgado que lo instruye), interesa saber cuál es el principal cargo (robo, homicidio, etc.) que se le imputa a un delincuente en cada delito en que está involucrado.

Uno o varios policías investigan cada uno de los casos.

Diseñe el esquema relacional. Indique en él las claves primarias, claves ajenas. Marque también con un asterisco los campos que aceptan valores nulos.

4. En la Universidad Nacional de Piura se desea llevar un control sobre los proyectos de investigación que se desarrollan. Para ello se decide emplear una base de datos que contenga toda la información sobre los proyectos, departamentos, grupos de investigación y docentes. Esta información se detalla a continuación:

Un departamento se identifica por su código y nombre y además pertenece a una Facultad de la universidad y tiene un jefe que es un docente de la universidad.

Dentro de un departamento se crean grupos de investigación. Cada grupo tiene un nombre único dentro del departamento (pero que puede ser el mismo en distintos departamentos) y está asociado a una línea de investigación (base de datos, inteligencia artificial, programación, etc.)

Cada grupo tiene un líder que ha de ser docente de la universidad.

Un docente está identificado por su DNI. De él se desea saber el nombre, titulación, años de experiencia en investigación, grupos de investigación en los que desarrolla o ha desarrollado su labor y proyectos en los que ha trabajado.

Cada proyecto de investigación tiene un nombre, un código único, un presupuesto, fechas de inicio y terminación y un grupo que lo desarrolla. Por otro lado, puede estar financiado por varios programas. Dentro de cada programa cada proyecto tiene un número asociado y una cantidad de dinero financiada (por ejemplo el proyecto “BDOO – Base de Datos Orientada a Objetos” tiene el número 3113 dentro del programa “Enfoque de Investigación” que le financia con 30,000 Nuevos Soles).

Un docente puede participar en varios proyectos. En cada proyecto se incorpora en una determinada fecha y cesa en otra, teniendo una determinada dedicación (en horas a la semana) durante ese periodo. Diseñe el esquema relacional. Indique en él las claves primarias, claves ajenas. Marque también con un asterisco los campos que aceptan valores nulos.

CAPÍTULO 6. ALGEBRA RELACIONAL Y LENGUAJE SQL

6.1. Algebra Relacional

El álgebra relacional se inspira en la teoría de conjuntos para especificar consultas en una base de datos relacional.

Para especificar una consulta en álgebra relacional, es preciso definir uno o más pasos que sirven para ir construyendo, mediante operaciones de álgebra relacional, una nueva relación que contenga los datos que respondan a la consulta a partir de las relaciones almacenadas. Los lenguajes basados en álgebra relacional son procedimentales, dado que los pasos que forman la consulta describen un procedimiento.

El álgebra relacional es un conjunto cerrado de operaciones que actúan sobre relaciones y producen relaciones como resultados. Pueden combinarse para construir expresiones más complejas.

Las operaciones del álgebra relacional se pueden clasificar en:

1. Operaciones Básicas o Primitivas: Son aquellas operaciones a partir de las cuales podemos definir el resto. Estas operaciones son la unión, la diferencia, el producto cartesiano, la selección y la proyección.
2. Operaciones Derivadas o no primitivas: Se pueden expresar a partir de las operaciones básicas. Estas operaciones son la intersección, Join, División y asociación.

6.1.1. Unión

La unión de dos relaciones R y S, es otra relación que contiene las tuplas que están en R, o en S, o en ambas, eliminándose las tuplas duplicadas.

R y S deben ser unión compatibles, es decir, definidas sobre el mismo conjunto de atributos.

Ejemplo: Sean las relaciones Empleado y Jefe:

Empleado			Jefe			Empleado U Jefe		
Codigo	Nombre	DNI	Codigo	Nombre	DNI	Codigo	Nombre	DNI
12	Pedro	02604087	18	Ana	02656784	12	Pedro	02604087
15	Miguel	71234567	22	Juan	02743568	15	Miguel	71234567
18	Ana	02656784				18	Ana	02656784
						22	Juan	02743568

6.1.2. Diferencia

La diferencia de dos relaciones R y S, es otra relación que contiene las tuplas que están en la relación R, pero no están en S.

Empleado

Codigo	Nombre	DNI
12	Pedro	02604087
15	Miguel	71234567
18	Ana	02656784

Jefe

Codigo	Nombre	DNI
18	Ana	02656784
22	Juan	02743568

Empleado - Jefe

Codigo	Nombre	DNI
12	Pedro	02604087
15	Miguel	71234567

Jefe - Empleado

Codigo	Nombre	DNI
22	Juan	02743568

6.1.3. Producto Cartesiano

Define una relación que es la concatenación de cada una de las tuplas de la relación R con cada una de las tuplas de la relación S.

Docente

Codigo	Nombre	Tipo
12	Pedro	PRI
15	Miguel	ASO

Proyecto

CodProy	Tiempo
PR1234	90
SZ4567	120

Docente X Proyecto

Codigo	Nombre	Tipo	CodProy	Tiempo
12	Pedro	PRI	PR1234	90
12	Pedro	PRI	SZ4567	120
15	Miguel	ASO	PR1234	90
15	Miguel	ASO	SZ4567	120

6.1.4. Selección

Define una relación con los mismos atributos que R y que contiene sólo aquellas filas de R que satisfacen la condición especificada (predicado):

$\sigma_{\text{predicado}}(R)$. Es una operación unaria.

En el siguiente ejemplo se seleccionan aquellos docentes cuyo sueldo es mayor a 3000.

Docente

Codigo	Nombre	Sueldo
12	Pedro	3000
15	Miguel	6200
18	Ana	3100
22	Martin	2800

 $\sigma_{\text{sueldo} > 3000}(\text{Docente})$

Codigo	Nombre	Sueldo
15	Miguel	6200
18	Ana	3100

6.1.5. Proyección

Define una relación que contiene un subconjunto vertical de R con los valores de los atributos especificados, eliminando tuplas duplicadas en el resultado:

$\pi_{\text{col1, col2, ..., col n}}(R)$. Es una operación unaria.

Docente

Codigo	Nombre	Sueldo
12	Pedro	3000
15	Miguel	6200
18	Ana	3000
22	Martin	2800

 $\pi_{\text{Nombre, Sueldo}}(\text{Docente})$

Nombre	Sueldo
Pedro	3000
Miguel	6200
Ana	3000
Martin	2800

 $\pi_{\text{Sueldo}}(\text{Docente})$

Sueldo
3000
6200
2800

6.1.6. Intersección

Define una relación que contiene el conjunto de todas las tuplas que están en la relación R como en S.

R y S deben ser unión compatibles.

Equivalencia con operadores básicos: $R \cap S = R - (R - S)$

Empleado

Codigo	Nombre	DNI
12	Pedro	02604087
15	Miguel	71234567
18	Ana	02656784

Jefe

Codigo	Nombre	DNI
18	Ana	02656784
22	Juan	02743568

Empleado \cap Jefe

Codigo	Nombre	DNI
18	Ana	02656784

6.1.7. Join**6.1.7.1. Join Natural**

El resultado es una relación con los atributos de ambas relaciones y se obtiene combinado las tuplas de ambas relaciones que tengan el mismo valor en los atributos comunes.

Normalmente la operación de join se realiza entre los atributos comunes de dos relaciones que corresponden a la clave primaria de una relación y la clave foránea (ajena) correspondiente de la otra relación.

6.1.7.2. Outer Join

Es una variante de join en la que se intenta mantener toda la información de los operandos, incluso para aquellas filas que no participan en el join.

Se rellenan con nulos las tuplas que no tienen correspondencia en el join.

Existen 3 variantes:

- a) Left Join: Se tienen en cuenta todas las tuplas del primer operando
- b) Right Join: Se tienen en cuenta todas las tuplas del segundo operando.
- c) Full Join: Se tienen en cuenta todas las tuplas de ambos operandos.

Alumno

Codigo	Nombre	CodCurso	CodCurso	NombreC
12	Pedro	SI2051	SI2051	Curso 1
15	Miguel	SG1345	SG1345	Curso 2
18	Ana	SS2347	SL2250	Curso 3

Curso**Alumno *_{LEFT} Curso**

Codigo	Nombre	CodCurso	NombreC
12	Pedro	SI2051	Curso 1
15	Miguel	SG1345	Curso 2
18	Ana	SS2347	NULL

Alumno *_{RIGHT} Curso

Codigo	Nombre	CodCurso	NombreC
12	Pedro	SI2051	Curso 1
15	Miguel	SG1345	Curso 2
NULL	NULL	SL2250	Curso 3

Alumno *_{FULL} Curso

Codigo	Nombre	CodCurso	NombreC
12	Pedro	SI2051	Curso 1
15	Miguel	SG1345	Curso 2
18	Ana	SS2347	NULL
NULL	NULL	SL2250	Curso 3

6.2. Lenguaje SQL

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por los diferentes motores de bases de datos para realizar

determinadas operaciones sobre los datos o sobre la estructura de los mismos. Pero como sucede con cualquier sistema de normalización hay excepciones para casi todo; de hecho, cada motor de bases de datos tiene sus peculiaridades y lo hace diferente de otro motor, por lo tanto, el lenguaje SQL normalizado (ANSI) no nos servirá para resolver todos los problemas, aunque si se puede asegurar que cualquier sentencia escrita en ANSI será interpretable por cualquier motor de datos.

6.2.1. *Comandos SQL*

Existen dos tipos de comandos SQL:

- Lenguaje de definición de datos (DDL), que permiten crear y definir nuevas bases de datos, campos e índices.
- Lenguaje de Manipulación de Datos (DML), que permiten actualizar, eliminar y consultar datos.

Los comandos DDL son:

CREATE: utilizado para crear nuevas tablas, campos e índices.

DROP: Empleado para eliminar tablas e índices.

ALTER: Se usa para modificar las tablas, agregando campos o cambiando su definición.

Los comandos DML son:

INSERT: Empleado para insertar datos en la tabla.

UPDATE: Se utiliza para modificar los valores de los campos y filas especificadas.

DELETE: Utilizado para eliminar registros de una tabla

SELECT: Se usa para consultar registros de una o varias tablas que satisfagan ciertas condiciones.

6.2.2. *Cláusulas*

Las cláusulas son condiciones de modificación utilizadas para definir los datos que se desea seleccionar o manipular.

FROM: Utilizada para especificar la tabla de la cual se van a seleccionar los registros.

WHERE: Utilizada para especificar las condiciones que deben reunir los registros que se van a seleccionar.

GROUP BY: Utilizada para separar los registros seleccionados en grupos específicos.

HAVING: Utilizada para expresar la condición que debe satisfacer cada grupo.

ORDER BY: Utilizada para ordenar los registros seleccionados de acuerdo a un orden específico.

Dada una sentencia SQL de selección que incluye todas las posibles cláusulas, el orden de ejecución de las mismas es el siguiente:

1. Cláusula FROM
2. Cláusula WHERE
3. Cláusula GROUP BY
4. Cláusula HAVING
5. Cláusula SELECT
6. Cláusula ORDER BY

Ejercicios Resueltos

1. Para el ejercicio 1 resuelto del capítulo 5. Desarrollar las siguientes consultas SQL:
 - a) Nombre de los peregrinos que han realizado el peregrinaje a Motupe y que recorrieron el camino 1

```
select distinct p.nombre
from Peregrino p
inner join PeregrinoLocalidad pl on p.idCarnet = pl.idCarnet
inner join EtapasLocalidades el on pl.idLocalidad = el.idLocalidad
inner join Etapas e on el.idEtapa = e.idEtapa
inner join Camino c on e.idCamino = c.idCamino
where c.nombre = 'Camino 1'
```
 - b) Nombre y Comunidad Autónoma de aquellas localidades que no posean albergues.

```
select nombre, comunidad
from Localidades
where idLocalidad not in (select idLocalidad from Albergue)
```
 - c) Nombre de la localidad y la cantidad de albergues que existen en dicha localidad

```
select l.nombre, count(*)
from Albergue a
inner join Localidad l on a.idLocalidad = l.idLocalidad
group by l.nombre
```
2. Para el ejercicio 2 resuelto del capítulo 5. Desarrollar las siguientes consultas SQL:
 - a) Nombre de todas las trabajadoras solteras.

```
select t.nombre
from Trabajador t
inner join EstadoCivil e on t.idEstCiv = e.idEstCiv
where t.sexo = 'F' and e.nombre = 'SOLTERO'
```

- b) Nombre de todos los trabajadores que son padres de familia y además la cantidad de hijos que tienen.
- ```
select t.nombre, count(*)
from Trabajador t
inner join TrabajadorFamiliar f on t.codTrab = f.codTrab
where f.tipo = 'H' --Hijos
group by t.nombre
having count(*) > 0
```
- c) La suma de los ingresos de cada trabajador en el mes de enero del 2013.
- ```
select t.nombre, sum(d.importe)
from Trabajador t
inner join Planilla p on t.codTrab = p.codTrab
inner join PlanillaDetalle d on p.idPlanilla = d.idPlanilla
inner join Conceptos c on d.idConcepto = c.idConcepto
where c.tipo = 'I'           --Ingresos
and p.mes = 1 and p.año = 2013
group by t.nombre
```

3. Para el ejercicio 3 resuelto del capítulo 5. Desarrollar las siguientes consultas SQL:

- a) La cantidad de puestos que existen en la URB. SANTA ISABEL
- ```
select l.nombre, count(*)
from Lugares l
inner join Puesto p on l.idLugar = p.idLugar
where l.nombre = 'URB. SANTA ISABEL'
group by l.nombre
```
- b) Los nombres de las personas que han rescindido su contrato
- ```
select distinct p.nombre
from Contrato c
inner join Persona p on c.idPersona = p.idPersona
where c.idContrato in (select idContrato from ContratoRescinde)
```
- c) El total de deuda que tiene cada persona por puesto en el año 2013
- ```
select p.nombre, d.idPuesto, sum(importe)
from Persona p
inner join Deuda d on p.idPersona = d.idPersona
where d.año = 2013
group by p.nombre, d.idPuesto
```

### ***Ejercicios Propuestos***

1. Para el ejercicio 1 propuesto del capítulo 5. Desarrollar las siguientes consultas SQL:
- a) La cantidad de especies que existen por grupo ornitológico.
- b) Los nombres de los observadores que han estado en una zona de observación del tipo laguna o río.

- c) Los nombres de asociaciones ornitológicas y la cantidad de trabajadores que tienen.
2. Para el ejercicio 2 propuesto del capítulo 5. Desarrollar las siguientes consultas SQL:
    - a) Nombre de los casos en los cuales ha concluido la investigación y la fecha del dictamen.
    - b) Nombre y DNI de los ciudadanos que se encuentran implicados en casos de corrupción y que pertenecen al partido político NUEVA PATRIA.
    - c) Nombre de los periódicos y la cantidad de casos de corrupción que han descubierto.
  3. Para el ejercicio 3 propuesto del capítulo 5. Desarrollar las siguientes consultas SQL:
    - a) Nombre de los policías que tienen una destreza de 8 en el uso de armas de la clase CORTO ALCANCE.
    - b) Los nombres de los policías que han efectuado arrestos y la cantidad de arrestos efectuados.
    - c) Los nombres de los delincuentes que han estado involucrados en algún caso con el principal cargo de homicidio.
  4. Para el ejercicio 3 propuesto del capítulo 5. Desarrollar las siguientes consultas SQL:
    - a) La relación de proyectos que son financiados por el programa "Enfoque de Investigación".
    - b) Los nombres de los grupos que están asociados a la línea de investigación "Inteligencia Artificial".
    - c) La cantidad de proyectos en los que ha intervenido cada docente de la facultad de Ingeniería Industrial.

## **BIBLIOGRAFÍA**

- CHARTE, De Miguel, Adoración; Piattini y Marcos. Diseño de base de datos relacionales. Alfa omega- Rama. 2000
- Kroenke, David. Procesamiento de bases de datos: Fundamentos, diseño e instrumentación. Prentice Hall Hispanoamericana. 5ta edición, 1996
- Korth H. F, Silberschatz. Fundamentos de Bases de Datos, Mc Graw Hill. 1993
- Hansen G. W.; Hansen J:V. Diseño y Administración de Bases de Dato. Prentice Hall. 1997
- Crovetto H., Christian. Oracle 9i. Megabyte. 1ra edición. 2004
- Pressman, Roger: Ingeniería de Software, Mc Graw Hill 2002. España.
- Stephen Buxton: Database Design Know It All. Morgan Kaufmann Publisher. 2009
- Silberschatz. Abraham: Fundamentos de Base de datos, Mc Graw Hill. 2002
- Adoración de Miguel Castaño: Diseño de Base de Datos-Problemas resueltos. Alfaomega. Ra-Ma, 2001