

El modelo de datos relacional

2

En esta unidad aprenderás a:

- 1 Describir la estructura del modelo de datos relacional.
- 2 Transformar el modelo E-R al modelo lógico relacional.
- 3 Diseñar bases de datos relacionales.
- 4 Normalizar esquemas relacionales.
- 5 Realizar operaciones básicas sobre tablas utilizando álgebra relacional.

2. El modelo de datos relacional

2.1 Introducción



2.1 Introducción

Ya se ha visto en la Unidad anterior lo que es un *sistema gestor de bases de datos*, y cuáles son sus objetivos. También se vio la *arquitectura ANSI* para su aplicación en la creación de las bases de datos. Se han estudiado distintos modelos conceptuales de datos pero no se ha llegado a realizar ningún diseño lógico. Esta Unidad trata del que actualmente es el principal modelo para las aplicaciones de procesamiento de datos, el *modelo relacional*, que presenta una forma muy simple y potente de representar los datos.

A lo largo de la Unidad, se exponen los fundamentos del modelo de datos relacional y su aplicación para el diseño lógico de datos y de bases de datos relacionales.

2.2 El modelo relacional

El modelo de datos relacional fue desarrollado por E.F. Codd para IBM, a finales de los años sesenta. Propone un modelo basado en la teoría matemática de las relaciones, con el objetivo de mantener la independencia de la estructura lógica respecto al modo de almacenamiento y otras características de tipo físico. El modelo de Codd persigue, al igual que la mayoría de los modelos de datos, los siguientes objetivos:

- **Independencia física de los datos.** El modo de almacenamiento de los datos no debe influir en su manipulación lógica.
- **Independencia lógica de los datos.** Los cambios que se realicen en los objetos de la base de datos no deben repercutir en los programas y usuarios que acceden a la misma.
- **Flexibilidad.** Para presentar a los usuarios los datos de la forma más adecuada a la aplicación que utilicen.
- **Uniformidad** en la presentación de las estructuras lógicas de los datos, que son tablas, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- **Sencillez.** Pues las características anteriores, así como unos lenguajes de usuario sencillos, hacen que este modelo sea fácil de comprender y utilizar por el usuario.

Para conseguir estos objetivos, Codd introduce el concepto de *relación (tabla)* como estructura básica del modelo. Todos los datos de una base de datos se representan en forma de relaciones cuyo contenido varía en el tiempo. El modelo relacional se basa en dos ramas de las matemáticas: la *teoría de conjuntos* y la *lógica de predicados*. Esto hace que sea un modelo seguro y robusto.

En 1985, Codd publica sus famosas doce reglas analizando algunos de los productos comerciales de la época, que debe cumplir cualquier base de datos para ser considerada relacional:

- **Regla de información.** Toda información de una base de datos relacional está representada mediante valores en tablas.



2. El modelo de datos relacional

2.2 El modelo relacional

- **Regla de acceso garantizado.** Se garantiza que todos los datos de una base relacional son lógicamente accesibles a través de una combinación de nombre de tabla, valor de clave primaria y nombre de columna.
- **Tratamiento sistemático de valores nulos.** Los valores nulos se soportan en los SGBD para representar la falta de información de un modo sistemático e independiente de los tipos de datos.
- **Catálogo en línea dinámico basado en el modelo relacional.** La descripción de la base de datos se representa en el ámbito lógico de la misma forma que los datos ordinarios, de modo que los usuarios autorizados pueden acceder a ellos utilizando el mismo lenguaje relacional.
- **Regla de sublenguaje completo de datos.** Un sistema relacional puede soportar varios lenguajes y varios modos de uso terminal. Sin embargo, debe haber al menos un lenguaje cuyas sentencias se puedan expresar mediante alguna sintaxis bien definida, como cadenas de caracteres, y que ofrezca completamente todos los puntos siguientes:
 - Definición de datos.
 - Definición de vistas.
 - Manipulación de datos (interactiva y por programa).
 - Restricciones de integridad.
 - Autorización.
 - Gestión de transacciones (comienzo, confirmación y vuelta atrás).
- **Regla de actualización de vista.** Todas las vistas, que sean teóricamente actualizables, son también actualizables por el sistema.
- **Inserción, actualización y supresión de alto nivel.** La capacidad de manejar una relación de base de datos o una relación derivada como un único operando se aplica no solamente a la recuperación de datos, sino también a la inserción, actualización y supresión de los datos.
- **Independencia física de los datos.** Los cambios que se efectúan tanto en la representación del almacenamiento, como en los métodos de acceso no deben afectar ni a los programas de aplicación ni a las actividades con los datos.
- **Independencia lógica de los datos.** Del mismo modo, los cambios que se efectúen sobre las tablas de la base de datos no modifican ni a los programas ni a las actividades con los datos.
- **Independencia de la integridad.** Las restricciones de integridad específicas de una base de datos relacional deben ser definidas mediante el sublenguaje de datos relacional y almacenarse en el catálogo de la base de datos.
- **Independencia de la distribución.** Un SGBD es independiente de la distribución.

2. El modelo de datos relacional

2.3 Estructura del modelo relacional



- **Regla de no subversión.** Si un SGBDR tiene un lenguaje de bajo nivel (una fila cada vez) no se puede utilizar para destruir o evitar las reglas de integridad o las restricciones expresadas en el lenguaje relacional de alto nivel (varias filas al mismo tiempo).

El **modelo relacional** propone una representación de la información que:

- Origine **esquemas** que representen fielmente la información, los objetos y las relaciones existentes entre ellos forman el dominio del problema.
- Sea **fácilmente entendida** por los usuarios.
- Sea **posible ampliar el esquema de la base de datos sin modificar la estructura lógica existente y los programas de aplicación.**
- Permite la **máxima flexibilidad** en la formulación de los interrogantes sobre la información mantenida en la base de datos.

2.3 Estructura del modelo relacional

Como ya se ha indicado anteriormente, la **relación** es el elemento básico del modelo relacional y se representa como una tabla, en la que se puede distinguir el nombre de la tabla, el conjunto de columnas que representan las **propiedades de la tabla** y que se les **llama atributos**, y el conjunto de filas llamadas **tuplas**, que contienen los valores que toma cada uno de los atributos para cada elemento de la relación.

Una **relación** tiene una serie de **elementos característicos** que la distinguen de una tabla:

- No admiten filas duplicadas.
- Las filas y columnas no están ordenadas.
- La tabla es plana. En el cruce de una fila y una columna sólo puede haber un valor, no se admiten atributos multivaluados.

En la figura que se muestra a continuación se representa una **relación** llamada **ALUMNO** en forma de tabla.

RELACIÓN ALUMNOS			
NUM_MAT	NOMBRE	APELLIDOS	CURSO
5467	JUAN	CABELLO	1BACH-A
3421	DOLORES	GARCÍA	1BACH-C
7622	JESÚS	SÁNCHEZ	2BACH-C

← ATRIBUTOS
← TUPLAS

Figura 2.1. Representación de una relación en forma de tabla.

A continuación, se exponen los elementos que constituyen el modelo relacional.



2. El modelo de datos relacional

2.3 Estructura del modelo relacional

A. Dominios y atributos

Se define **dominio** como el conjunto finito de valores homogéneos (todos del mismo tipo) y atómicos (son indivisibles), que puede tomar cada atributo. Los valores contenidos en una columna pertenecen a un dominio que previamente se ha definido. Todos los dominios tienen un nombre y un tipo de datos asociado. Existen dos tipos de dominios:

- **Dominios generales.** Son aquellos cuyos valores están comprendidos entre un máximo y un mínimo. Por ejemplo, el Código_postal, que está formado por todos los números enteros positivos de 5 cifras.
- **Dominios restringidos.** Son los que pertenecen a un conjunto de valores específico. Por ejemplo, Sexo, que puede tomar los valores H o M.

Se define **atributo** como el papel o rol que desempeña un dominio en una relación. Representa el uso de un dominio para una determinada relación. El atributo aporta un significado semántico a un dominio. Por ejemplo, en la relación ALUMNO podemos considerar los siguientes dominios:

- Atributo NUM_MAT, dominio: conjunto de enteros formados por 4 dígitos.
- Atributo NOMBRE, dominio: conjunto de 15 caracteres.
- Atributo APELLIDOS, dominio: conjunto de 20 caracteres.
- Atributo CURSO, dominio: conjunto de 7 caracteres.

B. Relaciones

La relación se representa mediante una tabla con filas y columnas. Un SGBD sólo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción sólo se aplica a la estructura lógica de la base de datos (nivel externo y conceptual de la arquitectura a tres niveles ANSI-SPARC). No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.

En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas a los campos o atributos de esos registros. La relación está formada por:

- **Atributo (columna).** Se trata de cada una de las columnas de la tabla. Las columnas tienen un nombre y pueden guardar un conjunto de valores. Una columna se identifica siempre por su nombre, nunca por su posición. El orden de las columnas en una tabla es irrelevante.
- **Tupla (fila).** Representa una fila de la tabla. En la Figura 2.2 aparece la tabla EMPLEADO con dos filas o tuplas.

2. El modelo de datos relacional

2.3 Estructura del modelo relacional



De las tablas se derivan los siguientes conceptos:

- **Cardinalidad.** Es el número de filas de la tabla. En el ejemplo anterior es dos.
- **Grado.** Es el número de columnas de la tabla. En el ejemplo anterior el grado es cinco.
- **Valor.** Viene representado por la intersección entre una fila y una columna. Por ejemplo, son valores de la tabla EMPLEADO: 13407877B, Milagros Suela Sarro, 1500.
- **Valor Null.** Representa la ausencia de información.

Nº Emple	Apellidos	Salario	Numdepart	FechaAlta
13407877B	Milagros Suela Sarro	1.500	10	18/11/90
41667891C	José María Cabello	2.000	20	29/10/92

Figura 2.2. Tabla EMPLEADO con dos filas.

◆ Definiciones formales

La definición matemática de una **relación entre n dominios** (D_1, D_2, \dots, D_n) es un subconjunto del producto cartesiano de estos dominios donde cada elemento de la relación, la tupla, es una serie de n valores ordenados. Una relación R definida sobre un conjunto de dominios D_1, D_2, \dots, D_n consta de:

- **Cabecera:** conjunto fijo de pares atributo:dominio, $\{(A_1: D_1), (A_2: D_2), \dots, (A_n: D_n)\}$, donde cada atributo corresponde a un único dominio y todos los atributos son distintos. El grado de la relación R es n .
- **Cuerpo:** conjunto variable de tuplas. Cada tupla es un conjunto de pares atributo: valor, $\{(A_1: v_{i1}), (A_2: v_{i2}), \dots, (A_n: v_{im})\}$, con $i=1, 2, \dots, m$, donde m representa la cardinalidad de la relación R . En cada par $(A_i: v_{ij})$ se tiene que el valor pertenece al dominio, $v_{ij} \in D_j$.

Para establecer la cabecera de la relación **EMPLEADO** primero se **definen los dominios**, que tienen un nombre y un tipo de datos asociado:

- Dominio **NUME_EMPL**, conjunto de 9 caracteres, de los cuales 8 son dígitos y el último es una letra.
- Dominio **NOMBRES**, conjunto de 25 caracteres.
- Dominio **PAGA**, conjunto de 4 dígitos.
- Dominio **DEPART**, posibles valores de números de departamento, rango de 01 a 99.
- Dominio **FECHAS**, conjunto de 10 caracteres.



2. El modelo de datos relacional

2.3 Estructura del modelo relacional

La **cabecera** de la tabla EMPLEADO sería:

```
{ (NºEmple:NUME_EMPLE), (Apellidos:NOMBRES), (Salario:PAGA)
, (Numdepart:DEPART), (FechaAlta:FECHAS) }
```

Una de las **tuplas** es:

```
{ (NºEmple: 13407877B ), (Apellidos: Milagros Suela Sarro),
(Salario: 1500), (Numdepart:10), (FechaAlta: 18/11/1990) }
```

◆ Propiedades de las relaciones

Las relaciones tienen las siguientes características:

- Cada **relación tiene un nombre y éste es distinto de los demás.**
- Los **valores de los atributos son atómicos**: en cada tupla cada atributo toma un **solo valor**. Se dice que las relaciones están **normalizadas**.
- No hay dos **atributos que se llamen igual**.
- El **orden de los atributos es irrelevante**, no están ordenados.
- Cada **tupla es distinta de las demás, no hay tuplas duplicadas**.
- Al igual que en los atributos, el **orden de las tuplas es irrelevante**, las tuplas no están ordenadas.

◆ Tipos de relaciones

En un SGBD relacional pueden existir varios tipos de relaciones, aunque no todos manejan todos los tipos. **Unas relaciones permanecen en la base de datos y otras son los resultados de consultas:**

- **Relaciones base.** Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada. Se corresponde con el nivel conceptual de la arquitectura ANSI.
- **Vistas.** Se corresponde con el nivel externo de la arquitectura ANSI. Son relaciones con nombre que se definen a partir de una consulta. No tienen datos almacenados, **lo que se almacena es la definición de la consulta. Se las llama también virtuales**.
- **Instantáneas.** Se corresponde con el nivel interno de la arquitectura ANSI. Son relaciones con nombre y derivadas de otras. Son relaciones de sólo lectura y se refrescan periódicamente por el sistema.
- **Resultados de consultas.** Son las resultantes de las consultas de usuario. No persisten en la base de datos.

2. El modelo de datos relacional

2.3 Estructura del modelo relacional



- **Resultados intermedios.** Son las relaciones que contienen los resultados de las subconsultas de usuario. No persisten en la base de datos.
- **Resultados temporales.** Son relaciones con nombre, similares a las relaciones base, pero se destruyen automáticamente en algún momento previamente determinado.

C. Claves

En una relación no hay tuplas repetidas. Se identifican de un modo único mediante los valores de sus atributos. Toda fila debe estar asociada con una clave que permita identificarla. A veces la fila se puede identificar por un único atributo, pero otras veces es necesario recurrir a más de un atributo. La clave debe cumplir dos requisitos:

- **Identificación única.** En cada fila de la tabla el valor de la clave ha de identificarla de forma única.
- **No redundancia.** No se puede descartar ningún atributo de la clave para identificar la fila.

Se define **clave candidata de una relación** como el conjunto de atributos que identifican única y mínimamente (necesarios para identificar la tupla) cada tupla de la relación. Siempre hay una clave candidata pues por definición no puede haber dos tuplas iguales. Habrá un **atributo o atributos que identifiquen la tupla**.

Una relación puede tener más de una clave candidata entre las cuales se distinguen:

- **Clave primaria o principal (primary key):** aquella clave candidata que el usuario escoge para identificar las tuplas de la relación. No puede tener valores nulos. Si sólo existe una clave candidata, ésta se elegirá como clave primaria.
- **Clave alternativa:** aquellas claves candidatas que no han sido escogidas como clave primaria.

Se denomina **clave ajena de una relación R1** al conjunto de atributos cuyos valores han de coincidir con los valores de la clave primaria de otra relación R2. Ambas claves estarán definidas sobre el mismo dominio y son muy importantes en el estudio de la integridad de datos del modelo relacional.

Caso práctico

- 1 Disponemos de las tablas TDEPART y TEMPLE. Las columnas de la tabla TDEPART son:

Nº. de departamento (NUMDEPT), Nombre de departamento (NOMDEPT), Presupuesto (PRESUPUESTO).

Las claves candidatas son Nº. de departamento y Nombre de departamento, pues son únicos y no se van a repetir. Podemos escoger como primaria el Nº. de departamento. Normalmente se elegirán como claves primarias los campos códigos o número de empleado, departamento o artículo, que suelen ser numéricos.

(Continúa)



2. El modelo de datos relacional

2.3 Estructura del modelo relacional

(Continuación)

Las columnas de la tabla TEMPLE son:

Nº. de empleado (NUMEMP), Apellido (APELIDO), Nº. de departamento (NUMDEP), Salario (SALARIO).

La clave candidata es el Nº. de empleado. El apellido se puede repetir, así que no se la considera candidata. Como sólo hay una se escoge primaria el Nº. de empleado.

El atributo Nº. de departamento es clave ajena; relaciona las tablas TEMPLE y TDEPART.

La Figura 2.3 muestra el contenido de las tablas TEMPLE y TDEPART con sus claves primarias y ajenas.

CLAVE PRIMARIA			
TDEPART	NUMDEPT	NOMDEPT	PRESUPUESTO
	D1	Marketing	1.000
	D2	Desarrollo	1.200
	D3	Investigación	5.000

CLAVE PRIMARIA		CLAVE AJENA	
TEMPLE	NUMEMP	APELLIDO	NUMDEP
	E1	López	D1
	E2	Fernández	D2
	E3	Martínez	D3
	E4	Sánchez	D2

Figura 2.3. Tablas TEMPLE y TDEPART con las claves primarias y ajenas.

D. Esquema de la base de datos

Una base de datos relacional es un conjunto de relaciones normalizadas. Para representar un esquema de una base de datos se debe dar el nombre de sus relaciones, los atributos de éstas, los dominios sobre los que se definen estos atributos, las claves primarias y las claves ajena.

En el esquema los nombres de las relaciones aparecen seguidos de los nombres de los atributos encerrados entre paréntesis. Las claves primarias son los atributos subrayados, y las claves ajena se representan mediante diagramas referenciales.

El esquema de la base de datos de empleados y departamentos es el siguiente:

TDEPART
TEMPLE
TEMPLE $\xrightarrow{\text{NUMDEP}}$ TEDEPART:
(NUMDEPT, NOMDEPT, PRESUPUESTO)
(NUMEMP, APELLIDO, NUMDEP, SALARIO)
Departamento al que pertenece el empleado

2. El modelo de datos relacional

2.4 Restricciones del modelo relacional



2.4 Restricciones del modelo relacional

En todos los modelos de datos existen restricciones que a la hora de diseñar una base de datos se tienen que tener en cuenta. Los datos almacenados en la base de datos han de adaptarse a las estructuras impuestas por el modelo y deben cumplir una serie de reglas para garantizar que son correctas. El modelo relacional impone dos tipos de restricciones. Algunas de ellas ya las hemos citado en las propiedades de las relaciones y las claves. Los tipos de restricciones son:

- **Restricciones inherentes al modelo**, indican las características propias de una relación que han de cumplirse obligatoriamente y que diferencian una relación de una tabla. No hay dos tuplas iguales, el orden de las tuplas y los atributos no es relevante. Cada atributo sólo puede tomar un único valor del dominio al que pertenece. Ningún atributo que forme parte de la clave primaria de una relación puede tomar un valor nulo.
- **Restricciones semánticas o de usuario**, que representan la semántica del mundo real. Éstas hacen que las ocurrencias de los esquemas de la base de datos sean válidos. Los mecanismos que proporciona el modelo para este tipo de restricciones son los siguientes:
 - a) La **restricción de clave primaria (PRIMARY KEY)** permite declarar uno o varios atributos como clave primaria de una relación.
 - b) La **restricción de unicidad (UNIQUE)** permite definir claves alternativas. Los valores de los atributos no pueden repetirse.
 - c) La **restricción de obligatoriedad (NOT NULL)** permite declarar si uno o varios atributos no pueden tomar valores nulos.
 - d) **Integridad referencial o restricción de clave ajena (FOREIGN KEY)**. Se utiliza para enlazar relaciones, mediante claves ajenas, de una base de datos. La integridad referencial indica que los valores de la clave ajena en la relación hijo se corresponden con los de la clave primaria en la relación padre.

Además de definir las claves ajenas hay que tener en cuenta las **operaciones de borrado y actualización** que se realizan sobre las tuplas de la relación referenciada. Las posibilidades son las siguientes:

- **Borrado y/o modificación en cascada (CASCADE)**. El borrado o modificación de una tupla en la relación padre (relación con la clave primaria) ocasiona un borrado o modificación de las tuplas relacionadas en la relación hija (relación que contiene la clave ajena). En el caso de empleados y departamentos, si se borra un departamento de la tabla TDEPART se borrarán los empleados que pertenecen a ese departamento. Igualmente ocurrirá si se modifica el NUMDEPT de la tabla TDEPART esa modificación se arrastra a los empleados que pertenezcan a ese departamento.
- **Borrado y/o modificación restringido (RESTRICT)**. En este caso no es posible realizar el borrado o la modificación de las tuplas de la relación padre si existen tuplas relacionadas en la relación hija. Es decir, no podría borrar un departamento que tiene empleados.



2. El modelo de datos relacional

2.4 Restricciones del modelo relacional

- **Borrado y/o modificación con puesta a nulos (SET NULL).** Esta restricción permite poner la clave ajena en la tabla referenciada a NULL si se produce el borrado o modificación en la tabla primaria o padre. Así pues, si se borra un departamento, a los empleados de ese departamento se les asignará NULL en el atributo NUMDEPT.
 - **Borrado y/o modificación con puesta a valor por defecto (SET DEFAULT).** En este caso, el valor que se pone en las claves ajena de la tabla referenciada es un valor por defecto que se habrá especificado en la creación de la tabla.
- e) **Restricciones de verificación (CHECK).** Esta restricción permite especificar condiciones que deban cumplir los valores de los atributos. Cada vez que se realice una inserción o una actualización de datos se comprueba si los valores cumplen la condición. Rechaza la operación si no se cumple.

A continuación, se muestran dos sentencias de creación de tablas, la tabla **FABRICANTES** (tabla maestra o padre) y la tabla **ARTÍCULOS** (tabla relacionada o hija). Un fabricante fabrica muchos artículos. Observa las restricciones:

```
CREATE TABLE FABRICANTES (
    CD_FAB NUMBER(3) NOT NULL DEFAULT 100,
    NOMBRE VARCHAR2(15) UNIQUE,
    PAIS VARCHAR2(15) CONSTRAINT CK_PA
        CHECK (PAIS=UPPER(PAIS)),
    CONSTRAINT PK_FA PRIMARY KEY (CD_FAB),
    CONSTRAINT CK_NO CHECK (NOMBRE=UPPER(NOMBRE))
) ;

CREATE TABLE ARTICULOS (
    ARTIC VARCHAR2(20) NOT NULL,
    COD_FA NUMBER(3) NOT NULL,
    PESO NUMBER(3) NOT NULL CONSTRAINT CK1_AR CHECK
        (PESO>0),
    CATEGORIA VARCHAR2(10) NOT NULL,
    PRECIO_VENTA NUMBER (4) CONSTRAINT CK2_AR CHECK
        (PRECIO_VENTA>0),
    PRECIO_COSTO NUMBER (4) CONSTRAINT CK3_AR CHECK
        (PRECIO_COSTO>0),
    EXISTENCIAS NUMBER (5),
    CONSTRAINT PK_ART PRIMARY KEY (ARTIC, COD_FA, PESO,
        CATEGORIA),
    CONSTRAINT FK_ARFA FOREIGN KEY (COD_FA) REFERENCES
        FABRICANTES (CD_FAB) ON DELETE CASCADE,
    CONSTRAINT CK_CAT CHECK (CATEGORIA
        IN('Primera', 'Segunda', 'Tercera'))
) ;
```

- f) **Aserciones (ASSERTION).** Son parecidas a la anterior, pero en este caso en lugar de afectar a una relación como CHECK, puede afectar a dos o más relaciones. La condición se establece sobre elementos de distintas relaciones. Pueden impli-

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



car a subconsultas en la condición. La definición de una aserción debe tener un nombre. Tiene vida por sí misma.

- g) **Disparadores (TRIGGER)**. Las restricciones anteriores son declarativas, sin embargo, este tipo es procedimental. El usuario podrá especificar una serie de acciones distintas ante una determinada condición. El usuario escribe el procedimiento a aplicar dependiendo del resultado de la condición. Los disparadores están soportados a partir de los estándares SQL3.

A continuación, se muestra un disparador de base de datos que audita las operaciones de inserción y borrado de datos en la tabla EMPLE. Cada vez que se realiza una operación de actualización o borrado se inserta en la tabla AUDITAREMPLE una fila que contendrá: la fecha y hora de la operación, el número y apellido del empleado afectado, y la operación que se realiza. La creación de triggers se verá en las siguientes unidades.

```
CREATE OR REPLACE TRIGGER auditar_act_emp
  BEFORE INSERT OR DELETE ON EMPLE FOR EACH ROW
BEGIN
  IF DELETING THEN
    INSERT INTO AUDITAREMPLE
      VALUES(TO_CHAR(sysdate, 'DD/MM/YY*HH24:MI*')
      || :OLD.EMP_NO || '*' || :OLD.APELLIDO || '*'
      BORRADO );
  ELSIF INSERTING THEN
    INSERT INTO AUDITAREMPLE
      VALUES(TO_CHAR(sysdate, 'DD/MM/YY*HH24:MI*')
      || :NEW.EMP_NO || '*' || :NEW.APELLIDO||'*' INSER-
      CION );
  END IF;
END;
```

2.5 Transformación de un esquema E-R a un esquema relacional

Una vez obtenido el esquema conceptual mediante el modelo E-R, hay que definir el **modelo lógico de datos**. Las reglas básicas para transformar un esquema conceptual E-R a un esquema relacional son las siguientes:

- Toda entidad se transforma en una tabla.
- Todo atributo se transforma en columna dentro de una tabla.
- El identificador único de la entidad se convierte en clave primaria.
- Toda relación N:M se transforma en una tabla que tendrá como clave primaria la concatenación de los atributos clave de las entidades que asocia.



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



Caso práctico

- 2 Dado el esquema que se muestra en la Figura 2.4, en el que se representan las compras de artículos que hacen los clientes, convierte el esquema E-R en relacional.

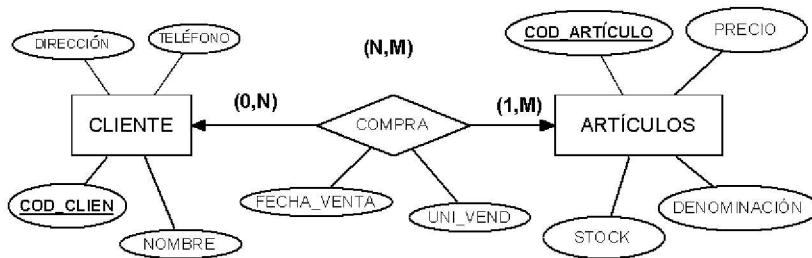


Figura 2.4. Esquema E-R CLIENTE-COMPRA-ARTICULOS.

1. Cada entidad se convierte en una tabla y los atributos de las entidades se convierten en columnas de las tablas. Lo representamos así:

CLIENTE (COD_CLIENTE, NOMBRE, DIRECCIÓN, TELÉFONO)
ARTICULOS (COD_ARTICULO, PRECIO, STOCK, DENOMINACIÓN)

2. La relación N:M se convierte a tabla. El nombre que se la da es el que pongamos a la relación, en este caso COMPRA. La clave estará formada por la concatenación de claves de las tablas anteriores. Estas a su vez pasan a ser claves ajena y referencian a las tablas CLIENTE y ARTÍCULO.

En esta relación, además de añadir las claves de las entidades anteriores, se añaden los atributos que intervienen en la relación. La tabla nos queda así:

COMPRA (COD_CLIENTE(FK), COD_ARTICULO(FK), UNI_VEND, FECHA_VENTA)

En la Figura 2.5 se muestra cómo quedaría gráficamente el esquema relacional creado en Access. Observa la ventana de relaciones en las que se indica las opciones de borrado y de modificación.

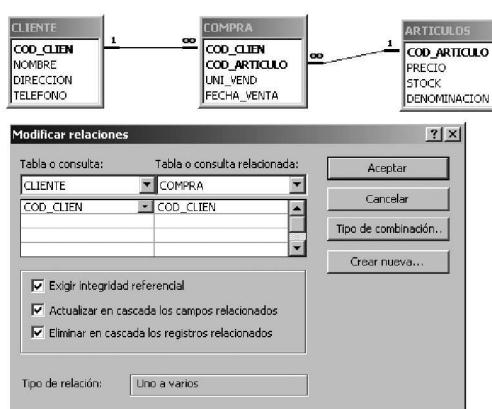


Figura 2.5. Representación gráfica relacional de CLIENTE-COMPRA-ARTÍCULOS.

(Continúa)

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



(Continuación)

Las opciones de borrado y de modificación suelen ser en cascada. A continuación, se muestra el código que crearía estas tres tablas:

```
CREATE TABLE CLIENTE (
    COD_CLIEN      NUMBER(6) NOT NULL PRIMARY KEY,
    NOMBRE          VARCHAR2(15),
    DIRECCION      VARCHAR2(15),
    TELEFONO       NUMBER(10));

CREATE TABLE ARTICULOS (
    COD_ARTICULO   NUMBER(6) NOT NULL PRIMARY KEY,
    PRECIO          NUMBER (6,2),
    STOCK           NUMBER (4),
    DENOMINACION   VARCHAR2(15));

CREATE TABLE COMPRA(
    COD_CLIEN      NUMBER(6) NOT NULL,
    COD_ARTICULO   NUMBER(6) NOT NULL,
    UNI_VEND       NUMBER (4),
    FECHA_VENTA   DATE,
    CONSTRAINT PK_COMPRA PRIMARY KEY (COD_CLIEN,COD_ARTICULO),
    CONSTRAINT FK_CLIEN FOREIGN KEY (COD_CLIEN)
        REFERENCES CLIENTE (COD_CLIEN)
        ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT FK_ARTIC FOREIGN KEY (COD_ARTICULO)
        REFERENCES ARTICULOS (COD_ARTICULO)
        ON DELETE CASCADE ON UPDATE CASCADE);
```

- En la transformación de relaciones 1:N existen dos soluciones:

- **Transformar la relación en una tabla.** Se hace como si se tratara de una relación N:M. Esta solución se realiza cuando se prevé que en un futuro la relación se convertirá en N:M y cuando la relación tiene atributos propios. También se crea una nueva tabla cuando la cardinalidad es opcional, es decir, (0,1) y (0,M). La clave de esta tabla es la de la entidad del lado *muchos*.

<-Participación

- **Propagar la clave.** Este caso se aplica cuando la cardinalidad es obligatoria, es decir, cuando tenemos cardinalidad (1,1) y (0,M) o (1,M). Se propaga el atributo principal de la entidad que tiene de cardinalidad máxima 1 a la que tiene de cardinalidad máxima N, desapareciendo el nombre de la relación. Si existen atributos propios en la relación, éstos también se propagarán.



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



Caso práctico

- 3 Dado el esquema que se muestra en la Figura 2.6, en el que se representa la clasificación de los libros en temas, convertir el esquema E-R en relacional.

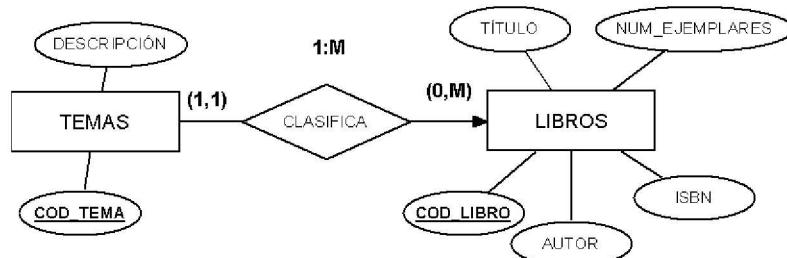


Figura 2.6. Esquema E-R. TEMA-CLASIFICA-LIBRO.

1. Se convierten a tabla las dos entidades:

TEMAS (COD_TEMA, DESCRIPCÓN)
LIBROS (COD_LIBRO, AUTOR, ISBN, TÍTULO, NUM_EJEMPLARES)

2. Se propaga la clave, de la entidad TEMAS a la entidad LIBROS. La entidad LIBROS quedará así:

LIBROS (COD_LIBRO, AUTOR, ISBN, TÍTULO, NUM_EJEMPLARES, COD_TEMA(FK))

En la Figura 2.7 podemos observar cómo quedaría gráficamente el esquema relacional, creado en Access.

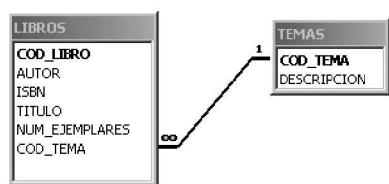


Figura 2.7. Representación gráfica relacional de TEMA-CLASIFICA-LIBRO.

- En la transformación de relaciones 1:1 se tienen en cuenta las cardinalidades de las entidades que participan. Existen dos soluciones:
 - **Transformar la relación en una tabla.** Si las entidades poseen cardinalidades (0,1), la relación se convierte en una tabla.
 - **Propagar la clave.** Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), conviene propagar la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad de cardinalidad (0,1). Si ambas entidades poseen cardinalidades (1,1), se puede propagar la clave de cualquiera de ellas a la tabla resultante de la otra. En este caso, también se pueden añadir los atributos de una entidad a otra, resultando una única tabla con todos los atributos de las entidades y de la relación, si los hubiera, eligiendo como clave primaria una de las dos.

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



Caso práctico

- 4 Consideramos la relación **EMPLEADO-OCUPA-PUESTOTRABAJO**. Un empleado ocupa un solo puesto de trabajo, y ese puesto de trabajo es ocupado por un solo empleado o por ninguno. El esquema E-R se muestra en la Figura 2.8; transformarlo al modelo relacional.

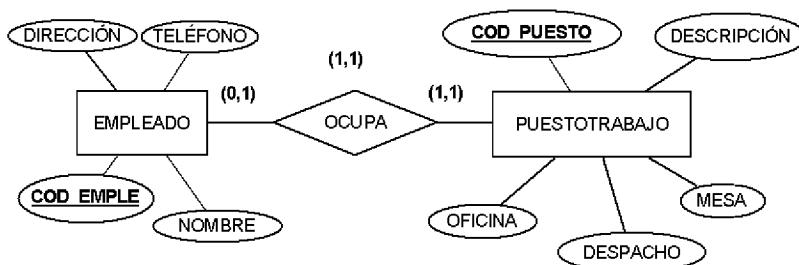


Figura 2.8. Esquema E-R de **EMPLEADO-OCUPA-PUESTOTRABAJO**.

En este caso, la clave se propaga desde la entidad PUESTOTRABAJO, con cardinalidad (1,1), a la entidad EMPLEADO, con cardinalidad (0,1). Las tablas las representaremos así:

PUESTOTRABAJO (COD_PUESTO, DESCRIPCIÓN, OFICINA, DESPACHO, MESA)
EMPLEADO (COD_EMPLE, NOMBRE, DIRECCIÓN, TELÉFONO, COD_PUESTO(FK))

En la Figura 2.9 se representa gráficamente el esquema relacional, creado en Access. En este caso, hay que tener en cuenta que el COD_PUESTO de EMPLEADO no se puede repetir, por lo tanto llevará la restricción de UNIQUE, aparte de ser clave ajena.



Figura 2.9. Representación gráfica relacional de **EMPLEADO-OCUPA-PUESTOTRABAJO**.

Actividades propuestas

- 1 Realiza el diagrama E-R que cumpla las especificaciones y pásalo al modelo de datos relacional.

Se desea mecanizar la biblioteca de un centro educativo. En la biblioteca existen fichas de autores y libros. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. Un libro está formado por ejemplares que son los que se prestan a los usuarios.

Así un libro tiene muchos ejemplares y un ejemplar pertenece sólo a un libro. De los ejemplares nos interesa saber la localización dentro de la biblioteca. Los ejemplares son prestados a los usuarios, un usuario puede tomar prestados varios ejemplares y un ejemplar puede ser prestado a varios usuarios. Del préstamo nos interesa saber la fecha de préstamo y la de devolución.



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional

A. Transformación de otros elementos del modelo E-R

• Relaciones reflexivas o recursivas

Son relaciones binarias en las que participa un tipo de entidad. En el proceso de convertir una relación reflexiva a tabla hay que tener en cuenta sobre todo la cardinalidad. Lo normal es que toda relación reflexiva se convierta en dos tablas, una para la entidad y otra para la relación. Se pueden presentar los siguientes casos:

- Si la relación es 1:1, la clave de la entidad se repite, con lo que la tabla resultante tendrá dos veces ese atributo, una como clave primaria y otra como clave ajena de ella misma. No se crea la segunda tabla.
- Si la relación es 1:M, podemos tener dos casos:
 - a) Caso de que la entidad muchos sea siempre obligatoria se procede como en el caso 1:1.
 - b) Si no es obligatoria, se crea una nueva tabla cuya clave será la de la entidad del lado muchos, y además se propaga la clave a la nueva tabla como clave ajena.



Caso práctico

- 5 Consideramos la relación EMPLEADO-DIRIGE-EMPLEADO. Un empleado puede dirigir a muchos empleados o a ninguno si es el que dirige. Y un empleado es dirigido por un director o por ninguno si el es el que dirige. En este caso no hay obligatoriedad en la entidad muchos. El esquema E-R se muestra en la figura 2.10, transformarlo al modelo relacional.

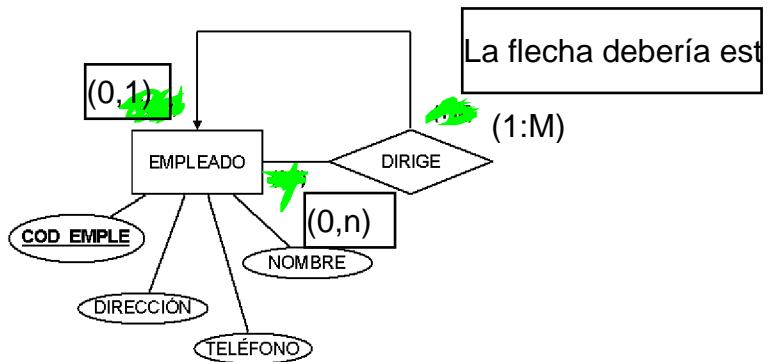


Figura 2.10. Esquema E-R de EMPLEADO-DIRIGE-EMPLEADO.

La tabla DIRIGE tiene como clave primaria el código de empleado, que a su vez será clave ajena. Además se le añade el código de empleado pero, en este caso, tendrá el papel de director, que a su vez será clave ajena a la tabla EMPLEADO. El resultado será el siguiente:

EMPLEADO (COD_EMPLE, DIRECCIÓN, TELÉFONO, NOMBRE)
DIRIGE (COD_EMPLE(FK), COD_DIREC(FK))

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



- Si es N:M, se trata igual que en las relaciones binarias. La tabla resultante de la relación contendrá dos veces la clave primaria de la entidad del lado muchos, más los atributos de la relación si los hubiera. La clave de esta nueva tabla será la combinación de las dos.

Caso práctico

- 6 Consideramos la relación: una pieza se compone de muchas piezas, que a su vez están compuestas de otras piezas, es decir, PIEZA-COMPONE-PIEZA. El esquema E-R se muestra en la Figura 2.11. Transfórmalo al modelo relacional.

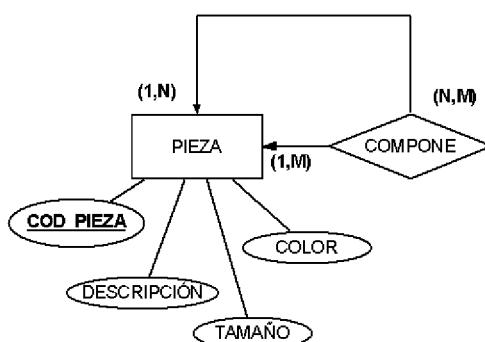


Figura 2.11. Esquema E-R de PIEZA-COMPONE-PIEZA.

En este caso, obtenemos la segunda tabla COMPONE_PIEZA en la que aparecerá repetido el código de pieza y formará la clave de la tabla. El atributo COD_PIEZ_CON representa la pieza que se compone de otras, y COD_PIEZA tiene el papel de pieza que compone a otras piezas. Ambos atributos son claves ajenas a la tabla PIEZA. La representación quedará así:

PIEZA	(<u>COD PIEZA</u> , DESCRIPCIÓN, TAMAÑO, COLOR)
COMPONE_PIEZA	(<u>COD PIEZ COM(FK)</u> , <u>COD PIEZA(FK)</u>)

- Generalizaciones, transformación de jerarquías al modelo relacional

El modelo relacional no dispone de mecanismos para la representación de las relaciones jerárquicas, así pues, las relaciones jerárquicas se tienen que eliminar. Para ello hay que tener en cuenta:

- La especialización que los subtipos tienen respecto a los supertipos, es decir, los atributos diferentes que tengan asociados cada uno de los subtipos, que son los que se diferencian con el resto de atributos de los otros subtipos.
- El tipo de especialización que representa el tipo de relación jerárquica: *total* o *parcial exclusiva*, y *total* o *parcial solapada*.
- Otros tipos de relación que mantengan tanto los subtipos como el supertipo.
- La forma en la que se va a acceder a la información que representan tanto el supertipo como el subtipo.



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional

Teniendo en cuenta los puntos señalados para pasar estas relaciones al modelo relacional se aplicará una de las siguientes reglas:

- A. **Integrar todas las entidades en una única eliminando a los subtipos.** Esta nueva entidad contendrá todos los atributos del supertipo, todos los de los subtipos, y los atributos discriminativos para distinguir a qué subentidad pertenece cada atributo. Todas las relaciones se mantienen con la nueva entidad. Esta regla puede aplicarse a cualquier tipo de jerarquía. **La gran ventaja es la simplicidad pues todo se reduce a una entidad.** El gran inconveniente es que se generan demasiados valores nulos en los atributos opcionales propios de cada entidad.
- B. **Eliminación del supertipo,** transfiriendo los atributos del supertipo a cada uno de los subtipos. Las relaciones del supertipo se consideran para cada uno de los subtipos. La clave genérica del supertipo pasa a cada uno de los subtipos. Sólo puede ser aplicada para jerarquías totales y exclusivas. Inconvenientes que presenta esta transformación:
 - Se crea redundancia en la información, pues los atributos del supertipo se repiten en cada uno de los subtipos.
 - El número de relaciones aumenta, pues si el supertipo tiene relaciones, éstas pasan a cada uno de los subtipos.
- C. **Insertar una relación 1:1 entre el supertipo y cada uno de los subtipos.** Los atributos se mantienen y cada subtipo se identificará con la clave ajena del supertipo. El supertipo mantendrá una relación 1:1 con cada subtipo. Los subtipos mantendrán, si la relación es exclusiva, la cardinalidad mínima 0, y si es solapada 0 ó 1.

Caso práctico

- 7 Consideraremos los profesores que imparten clases en dos tipos de centros educativos: públicos y privados. Un profesor puede impartir clase en varios centros, ya sean públicos o privados. Consideramos la asignatura como atributo de la relación entre profesor imparte en centro. Los centros educativos sólo pueden ser de estos dos tipos. Un centro público no puede ser a la vez privado. Los atributos específicos para los centros públicos son el presupuesto y los servicios, para los privados la organización y la cuota. La jerarquía a representar es total y exclusiva. En la Figura 2.12 se presenta el esquema E-R.

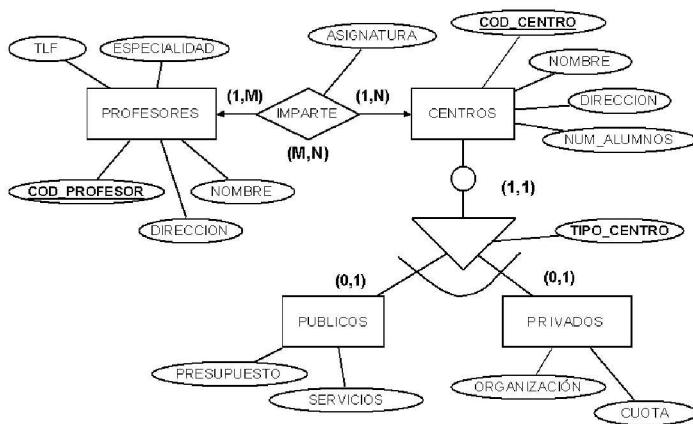


Figura 2.12. Esquema E-R de PROFESORES-CENTROS.

(Continúa)

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



(Continuación)

Observa que se ha añadido el atributo TIPO_CENTRO que no se ha dado en el enunciado, sin embargo en las generalizaciones conviene añadir un atributo que identifique el tipo del subtipo. Para la transformación podemos aplicar cualquiera de las tres reglas ya que se trata de una jerarquía total y exclusiva:

- A. Integraremos los subtipos en el supertipo. El esquema E-R será el que se muestra en la Figura 2.13:

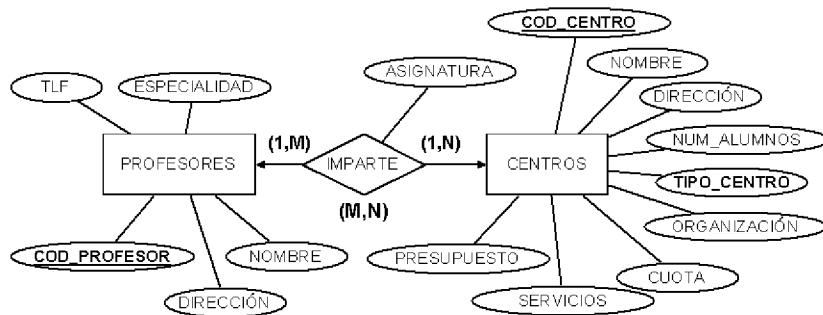


Figura 2.13. Solución A. Una única entidad.

El resultado en el modelo relacional contará con las siguientes tablas:

PROFESORES (COD PROFESOR, DIRECCIÓN, NOMBRE, TLF, ESPECIALIDAD)
 CENTROS (COD CENTRO, NOMBRE, DIRECCIÓN, NUM_ALUMNOS, TIPO_CENTRO, ORGANIZACIÓN, CUOTA,
 SERVICIOS, PRESUPUESTO)
 IMPARTE (COD PROFESOR(FK), COD CENTRO(FK), ASIGNATURA)

- B. Eliminación del supertipo, el esquema E-R se muestra en la Figura 2.14:

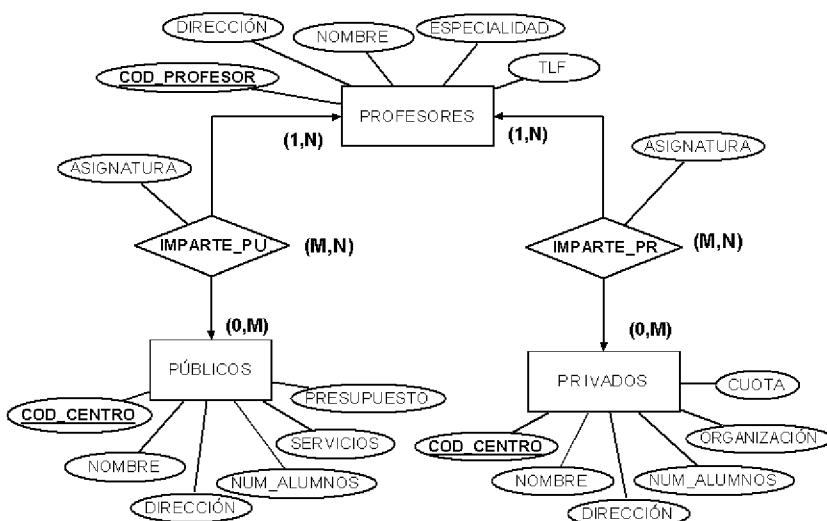


Figura 2.14. Solución B. Eliminación del supertipo, dos entidades.

(Continúa)



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional

(Continuación)

El **TIPO_CENTRO** se puede arrastrar a las subentidades pero no tendría mucho sentido, pues tomaría siempre el mismo valor, es decir, en un **centro público** sería siempre **público** y en uno **concertado**, **concertado**. Así pues se elimina. El resultado en el modelo relacional contará con las siguientes tablas:

PROFESORES	(<u>COD_PROFESOR</u> , DIRECCIÓN, NOMBRE, TLF, ESPECIALIDAD)
PÚBLICOS	(<u>COD_CENTRO</u> , NOMBRE, DIRECCIÓN, NUM_ALUMNOS, SERVICIOS, PRESUPUESTO)
PRIVADOS	(<u>COD_CENTRO</u> , NOMBRE, DIRECCIÓN, NUM_ALUMNOS, ORGANIZACIÓN, CUOTA)
IMPARTE_PR	(<u>COD_PROFESOR(FK)</u> , <u>COD_CENTRO(FK)</u> , ASIGNATURA)
IMPARTE_PU	(<u>COD_PROFESOR(FK)</u> , <u>COD_CENTRO(FK)</u> , ASIGNATURA)

AMBAS FK SON PK

Si nos fijamos en el esquema relacional resultante vemos que las tablas PÚBLICOS y PRIVADOS tiene la misma clave, y no existe ningún condicionante lógico que impida que un centro esté en las dos tablas, algo que va en contra del enunciado del problema. **Para controlar esta situación habrá que recurrir a los programas que manejen las tablas o a la creación de triggers asociados a cada una de las tablas.**

Del mismo modo, podemos pensar que las tablas IMPARTE_PR e IMPARTE_PU, son iguales y podrían agruparse en una sola, sin embargo, habría problemas con el control de la integridad puesto que habría que definir la clave de esta tabla como clave ajena de las tablas PÚBLICOS y PRIVADOS, y esto no sería correcto.

C. Inserta una relación 1:1 entre el supertipo y cada uno de los subtipos. El esquema E-R se muestra en la Figura 2.15:

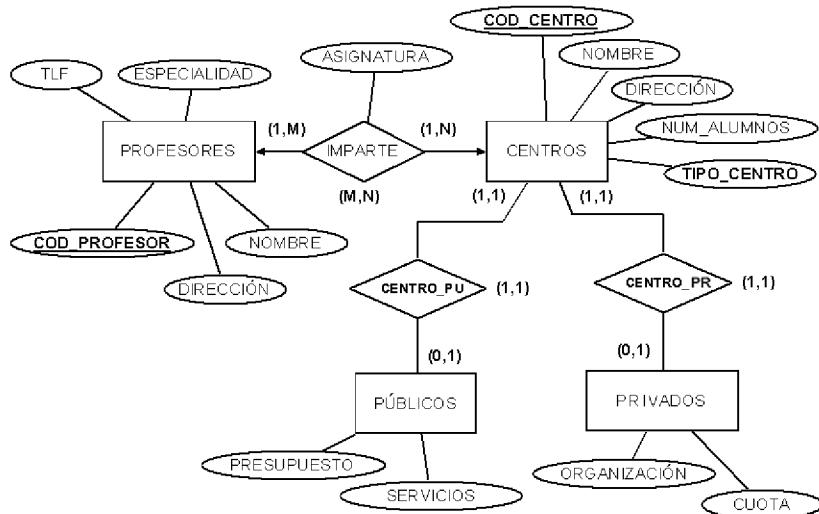


Figura 2.15. Solución C. Relación 1:1 entre supertipo y subtipos.

El atributo TIPO_CENTRO se incluye en el supertipo. El resultado en el modelo relacional contará con las siguientes tablas:

PROFESORES	(<u>COD_PROFESOR</u> , DIRECCION, NOMBRE, TLF, ESPECIALIDAD)
CENTROS	(<u>COD_CENTRO</u> , NOMBRE, DIRECCION, NUM_ALUMNOS, TIPO_CENTRO)
PÚBLICOS	(<u>COD_CENTRO(FK)</u> , SERVICIOS, PRESUPUESTO)
PRIVADOS	(<u>COD_CENTRO(FK)</u> , ORGANIZACIÓN, CUOTA)
IMPARTE	(<u>COD_PROFESOR(FK)</u> , <u>COD_CENTRO(FK)</u> , ASIGNATURA)

2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



- **Relaciones N-arias**

En este tipo de relaciones se agrupan 3 o más entidades, y para pasar al modelo de datos relacional cada entidad se convierte en tabla, así como la relación, que va a contener los atributos propios de ella más las claves de todas las entidades. La clave de la tabla resultante será la concatenación de las claves de las entidades. Hay que tener en cuenta:

- Si la relación es M:M:M, es decir, si todas las entidades participan con cardinalidad máxima M, la clave de la tabla resultante es la unión de las claves de las entidades que relaciona. Esa tabla incluirá los atributos de la relación si los hubiera.

Caso práctico

- 8 Suponemos una relación ternaria entre las entidades PROFESORES-CURSOS-ASIGNATURAS, en la que un profesor imparte en varios cursos varias asignaturas, y además las asignaturas son impartidas por varios profesores en varios cursos. El esquema E-R se muestra en la Figura 2.16. Transformarlo al modelo relacional.

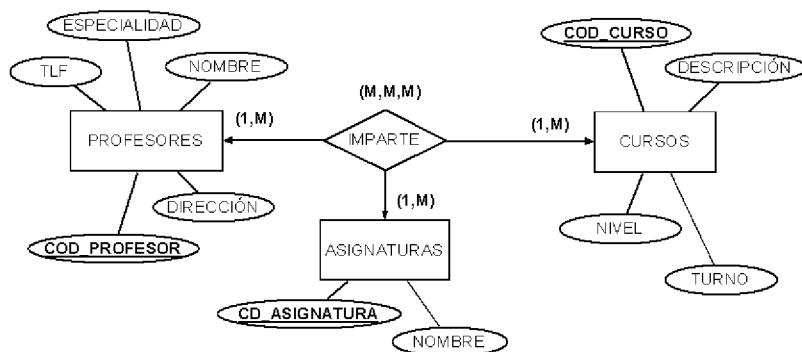


Figura 2.16. Esquema E-R de PROFESORES-CURSOS-ASIGNATURAS.

El resultado en el modelo relacional contará con las siguientes tablas:

PROFESORES	(<u>COD PROFESOR</u> , DIRECCIÓN, NOMBRE, TLF, ESPECIALIDAD)
CURSOS	(<u>COD CURSO</u> , DESCRIPCIÓN, NIVEL, TURNO)
ASIGNATURAS	(<u>COD_ASIGNATURA</u> , NOMBRE)
IMPARTE	(<u>COD PROFESOR(FK)</u> , <u>COD_CURSO(FK)</u> , <u>COD_ASIGNATURA(FK)</u>)

- Si la relación es 1:M:M, es decir, una de las entidades participa con cardinalidad máxima 1, la clave de esta entidad no pasa a formar parte de la clave de la tabla resultante, pero forma parte de la relación como un atributo más.



2. El modelo de datos relacional

2.5 Transformación de un esquema E-R a un esquema relacional



Caso práctico

- 9 Suponemos el caso de una tienda de venta de coches, en la que un empleado vende muchos coches a muchos clientes, y los coches son vendidos por un solo empleado. En la venta hay que tener en cuenta la forma de pago y la fecha de venta. El esquema E-R se muestra en la Figura 2.17. Transfórmalo al modelo relacional.

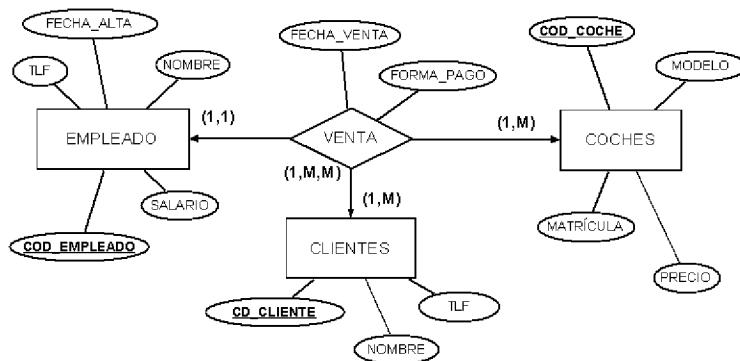


Figura 2.17. Esquema E-R de EMPLEADO-CLIENTES-COCHES.

El resultado en el modelo relacional es:

CLIENTES	(<u>COD_CLIENTE</u> , NOMBRE, TLF)
EMPLEADO	(<u>COD_EMPLEADO</u> , NOMBRE, TLF, SALARIO, FECHA_ALTA)
COCHES	(<u>COD_COCHE</u> , MATRICULA, MODELO, PRECIO)
VENTA	(<u>COD_COCHE(FK)</u> , <u>COD_CLIENTE(FK)</u> , COD_EMPLEADO, FORMA_PAGO, FECHA_VENTA)

En la Tabla 2.1 se muestra un resumen de las transformaciones E-R a Relacional:

Modelo Entidad-Relación (Chen)		Modelo relacional	
Entidad		Tabla	
Binarias	1:N	Propagar la clave: clave ajena (tabla con N), o bien otra tabla.	
	1:1	Propagar la clave: clave ajena, o bien otra tabla o bien uniendo las dos entidades en una tabla.	
	M:N	Otra tabla nueva, con clave primaria igual a la suma de las claves primarias de las dos tablas.	
Reflexivas	1:N	Clave ajena o bien otra tabla.	
	1:1	Clave ajena o bien otra tabla.	
	M:N	Otra tabla nueva, relación con clave principal igual a la clave principal de la entidad duplicada.	
Ternarias		<p>Se crea una tabla nueva y a la hora de elegir la clave se tendrá en cuenta:</p> <ol style="list-style-type: none">Concatenación de claves primarias de las entidades, con grado diferente a 1 (M,M,M...)Si alguna tiene cardinalidad máxima 1, al menos ha de haber (N-1) claves primarias de otras (N-1) entidades, y han de participar en la relación las claves primarias de las entidades con cardinalidad máxima 1.	

Tabla 2.1. Resumen de la conversión del modelo E-R al relacional.

2. El modelo de datos relacional

2.6 Pérdida de semántica en la transformación al modelo relacional



2.6 Pérdida de semántica en la transformación al modelo relacional

Algunas restricciones son necesarias controlarlas con mecanismos externos al modelo relacional dado que muchos de SGBD no implementan el modelo relacional completo. Por ejemplo, los mecanismos tipo CHECK y ASSERTION no suelen estar disponibles en todos los SGBD lo que implica que hay que recurrir a otros medios para recoger estas restricciones como: el uso de disparadores, procedimientos almacenados o aplicaciones externas.

Algunas de las restricciones de los esquemas E-R que deben contemplarse en la transformación al modelo relacional mediante *checks*, aserciones o disparadores son:

- Cardinalidades mínimas de 1 en relaciones N:M y 1:N (excluyendo aquellas que se controlan con la restricción NOT NULL cuando se realiza una propagación de clave).
- Cardinalidades máximas conocidas en relaciones binarias N:M y 1:N y relaciones ternarias.
- Exclusividad en las generalizaciones.
- Inserciones y borrado en las generalizaciones.
- Restricciones que no figuran en el enunciado original pero que se consideran adecuadas o convenientes (por ejemplo, restricciones que implican operadores de comparación de fechas, etcétera).

2.7 Normalización de esquemas relationales

La **normalización** es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad. Aquí no se utilizará la normalización como una técnica de diseño de bases de datos, sino como una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias no deseadas entre atributos. Las ventajas de la normalización son las siguientes:

- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.

Uno de los conceptos fundamentales en la normalización es el de **dependencia funcional**.

Una **dependencia funcional** es una relación entre atributos de una misma relación (tabla).



2. El modelo de datos relacional

2.7 Normalización de esquemas relationales

Si X e Y son atributos de la relación R , se dice que Y es funcionalmente dependiente de X (se denota por $X \rightarrow Y$) si cada valor de X tiene asociado un solo valor de Y (X e Y pueden constar de uno o varios atributos). A X se le denomina *determinante*, ya que X determina el valor de Y . Se dice que el atributo Y es *completamente dependiente* de X si depende funcionalmente de X y no depende de ningún subconjunto de X .

Esto es lo mismo que decir que si dos tuplas de R tienen el mismo valor para su atributo X , forzosamente han de tener el mismo valor para el atributo Y .

La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos no lo determina una serie abstracta de reglas, sino, más bien, los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumple una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplen.

La normalización se lleva a cabo en una serie pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal.

A. Cálculo de dependencias

Las **dependencias** son propiedades inherentes al contenido semántico de los datos formando parte de las restricciones de usuario del modelo relacional. Entre los atributos de una relación pueden existir dependencias de varios tipos:

- **Dependencias funcionales**

Son de primordial importancia a la hora de encontrar y eliminar la redundancia de los datos almacenados en las tablas de una base de datos relacional, se centran en el estudio de las dependencias que presenta cada atributo de una relación con respecto al resto de atributos de la misma.

Dada una relación R que contiene los atributos X e Y se dice que Y depende funcionalmente de X ($X \rightarrow Y$) si y sólo si en todo momento cada valor de X tiene asociado un solo valor de Y . Esto es lo mismo que decir que si dos tuplas de R tienen el mismo valor para su atributo X forzosamente han de tener el mismo valor para el atributo Y .

Tipos de dependencias:

- **Dependencias completa y parcial.** Dado un atributo compuesto X formado por los atributos X_1 y X_2 , se dice que el atributo Y tiene una dependencia funcional completa con respecto a X si: $X_1 \rightarrow Y$; $X_2 \rightarrow Y$; $Y -/ \rightarrow X$ (Y no implica X) pero $X \rightarrow Y$.

2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



Por el contrario, dado un atributo compuesto X formado por los atributos X1 y X2, se dice que el atributo Y tiene una dependencia funcional parcial con respecto a X si: $X_1 \rightarrow Y$ o $X_2 \rightarrow Y$ y $X \rightarrow Y$

- **Dependencia transitiva.** Dados los atributos X, Y y Z de la relación R en la que existen las siguientes dependencias funcionales $X \rightarrow Y$; $Y \rightarrow Z$; se dice que Z tiene una dependencia transitiva respecto a X a través de Y: $X \rightarrow Z$. Por ejemplo: Nombre de alumno Dirección; Dirección Ciudad; Nombre de alumno Ciudad.
- **Dependencia multivaluada o de valores múltiples.** Sean X e Y dos descriptores, X multidetermina a Y ($X \rightarrow Y$) si para cada valor de X existe un conjunto bien definido de valores posibles en Y, con independencia del resto de los atributos de la relación. Este tipo de dependencias produce redundancia de datos, como se puede apreciar en la Tabla 2.2, en donde las claves 34567-B y 89345-M tienen dos registros para mantener la serie de datos en forma independiente. Esto ocasiona que al realizarse una actualización se requiera de demasiadas operaciones para tal fin.

En la tabla siguiente, el atributo DNI multidetermina a Titulación; ver Tabla 2.2:

DNI	TITULACIÓN
34567-B	Licenciado en Exactas
34567-B	Licenciado en Ciencias Físicas
56667-C	Ingeniero de Telecomunicaciones
89345-M	Licenciado en Historia
89345-M	Licenciado en Filosofía

Tabla 2.2. Dependencia multivaluada.

- **Dependencias de reunión o en combinación.** Se dice que una relación satisface la dependencia de reunión (X, Y, \dots, Z) si y sólo si la relación es igual a la reunión de sus proyecciones según X, Y, \dots, Z . Donde X, Y, \dots, Z son subconjuntos del conjunto de atributos de la relación.

- **Reglas de normalización**

Se dice que una relación está en una forma normal si satisface un cierto conjunto específico de restricciones impuestas por la regla de normalización correspondiente. La aplicación de una regla es una operación que toma como entrada una relación y da como resultado dos o más relaciones.

- **Primera forma normal. 1FN**

Se dice que una relación está en 1FN si y sólo si los valores que componen cada atributo de una tupla son atómicos, es decir, cada atributo de la relación toma un único valor del dominio correspondiente, o lo que es lo mismo no existen grupos repetitivos.

La tabla estará en 1FN si tiene un solo valor en la intersección de cada fila con cada columna. Un conjunto de relaciones se encuentra en 1FN si ninguna de ellas tiene grupos repetitivos.



2. El modelo de datos relacional

2.7 Normalización de esquemas relationales

Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación. Hay dos formas de eliminar los grupos repetitivos:

- Repetir los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales.
- La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.



Caso práctico

- ⑩ La aplicación de esta regla es fácil. Este proceso se realiza casi siempre en el análisis del problema. Por ejemplo, consideramos la tabla ALUMNO, con clave primaria COD_ALUMNO, en la que el atributo TLF puede tomar varios valores: el móvil, el de casa, el del padre, el de la madre, etcétera. Ver Tabla 2.3:

COD_ALUMNO	NOMBRE	APELLIDO	TLF	DIRECCION
1111	PEPE	GARCÍA	678-900800 91-2233441 91-1231232	C/Las cañas 45
2222	MARÍA	SUÁREZ	91-7008001	C/Mayor 12
3333	JUAN	GIL	91-7562324 660-111222	C/La plaza
4444	FRANCISCO	MONTOYA	678-556443	C/La arboleda

Tabla 2.3. Tabla ALUMNO que no está en 1FN.

Esta tabla no está en 1FN, ya que hay dos alumnos que tienen varios teléfonos. Para que esta tabla esté en 1FN se puede:

- Definir como clave primaria de la tabla COD_ALUMNO, y el TLF, con el fin de que cada atributo tome un único valor en la tupla correspondiente. Ver Tabla 2.4.

COD_ALUMNO	TLF	NOMBRE	APELLIDO	DIRECCIÓN
1111	678-900800	PEPE	GARCÍA	C/Las cañas 45
1111	91-2233441	PEPE	GARCÍA	C/Las cañas 45
1111	91-1231232	PEPE	GARCÍA	C/Las cañas 45
2222	91-7008001	MARÍA	SUÁREZ	C/Mayor 12
3333	91-7562324	JUAN	GIL	C/La plaza
3333	660-111222	JUAN	GIL	C/La plaza
4444	678-556443	FRANCISCO	MONTOYA	C/La arboleda

Tabla 2.4. Tabla ALUMNO, primera transformación a 1FN.

(Continúa)

2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



(Continuación)

- O también se eliminan los grupos repetitivos (TLF) y se crea una relación (tabla) junto con la clave inicial. Ver Tablas 2.5 y 2.6.

COD ALUMNO	NOMBRE	APELLIDO	DIRECCION
1111	PEPE	GARCÍA	C/Las cañas 45
2222	MARÍA	SUÁREZ	C/Mayor 12
3333	JUAN	GIL	C/La plaza
4444	FRANCISCO	MONToya	C/La arboleda

Tabla 2.5. Tabla ALUMNO segunda transformación a 1FN.

La segunda tabla estará formada por el TLF, y el COD_ALUMNO. Los dos formarán la clave. Ver Tabla 2.6:

COD ALUMNO(FK)	TLF
1111	678-900800
1111	91-2233441
1111	91-1231232
2222	91-7008001

Tabla 2.6. Tabla ALUM_TLF en 1FN.

• Segunda forma normal. 2FN

Se dice que una relación se encuentra en 2FN si y sólo si satisface la 1FN, y cada atributo de la relación que no está en la clave depende funcionalmente de forma completa de la clave primaria de la relación. La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos.

Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.

Si la clave primaria está formada por un solo atributo y la relación está en 1FN, ya está en 2FN.

Teorema de la 2FN: sea una relación formada por los atributos A, B, C, D con clave primaria compuesta por los atributos A y B. Si se cumple que (D depende funcionalmente de A): $A \rightarrow D$, entonces la relación puede descomponerse en dos relaciones R1 y R2 con los atributos respectivos: R1 (A, D) y R2 (A, B, C).

Para pasar una relación en 1FN a 2FN hay que eliminar las dependencias parciales de la clave primaria. Para ello, se eliminan los atributos, que son funcionalmente dependientes, y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).

Se crearán dos tablas para eliminar las dependencias funcionales, una de ellas tendrá los atributos que dependen funcionalmente de la clave, y la otra los atributos que forman parte de la clave de la que dependen.



2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



Caso práctico

- 11** Supongamos que tenemos una relación ALUMNO en la que representamos los datos de los alumnos y las notas en cada una de las asignaturas en que está matriculado. La clave es el número de matrícula COD_ALUMNO y la asignatura ASIGNATURA, ver Tabla 2.7:

COD_ALUMNO	NOM_ALUM	APE_ALUM	ASIGNATURA	NOTA	CURSO	AULA
1111	PEPE	GARCÍA	LENGUA I	5	1	15
1111	PEPE	GARCÍA	IDIOMA	5	2	16
2222	MARÍA	SUAREZ	IDIOMA	7	2	16
2222	MARÍA	SUAREZ	CIENCIAS	7	2	14
3333	JUAN	GIL	PLÁSTICA	6	1	18
3333	JUAN	GIL	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONToya	LENGUA II	4	2	11
4444	FRANCISCO	MONToya	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONToya	CIENCIAS	8	1	14

Tabla 2.7. Relación ALUMNO para transformar a 2FN.

Es obvio que todos los atributos no dependen de la clave completa (COD_ALUMNO, ASIGNATURA). En primer lugar, hay que ver las dependencias funcionales de cada uno de los atributos con respecto a los atributos de la clave y el resto de atributos:

- NOM_ALUM y APE_ALUM, sólo dependen de COD_ALUMNO.
- Los atributos CURSO y AULA están relacionados con la ASIGNATURA, es decir, existe una dependencia entre ASIGNATURA → CURSO, ASIGNATURA → AULA. Una asignatura pertenece a un curso y se imparte en un aula.
- El atributo NOTA depende funcionalmente de la clave, pues para que haya una nota tiene que haber una asignatura y un alumno.

Vistas las dependencias funcionales llegamos a la siguiente conclusión para que la relación ALUMNO esté en 2FN necesitamos crear tres relaciones: ALUMNO, ASIGNATURAS y NOTAS (ver Tablas 2.8, 2.9 y 2.10):

COD_ALUMNO	NOM_ALUM	APE_ALUM
1111	PEPE	GARCÍA
2222	MARÍA	SUAREZ
3333	JUAN	GIL
4444	FRANCISCO	MONToya

Tabla 2.8. Tabla ALUMNO en 2FN.

ASIGNATURA	CURSO	AULA
LENGUA I	1	15
IDIOMA	2	16
CIENCIAS	2	14
PLÁSTICA	1	18
MATEMÁTICAS I	1	12
LENGUA II	2	11

Tabla 2.9. Tabla ASIGNATURAS en 2FN.

COD_ALUMNO (FK)	ASIGNATURA (FK)	NOTA
1111	LENGUA I	5
1111	IDIOMA	5
2222	IDIOMA	7
2222	CIENCIAS	7
3333	PLÁSTICA	6
3333	MATEMÁTICAS I	6
4444	LENGUA II	4
4444	MATEMÁTICAS I	6
4444	CIENCIAS	8

Tabla 2.10. Tabla NOTAS en 2FN.

2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



- **Tercera forma normal. 3FN**

Una relación está en tercera forma normal si, y sólo si, está en 2FN y, además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. Es decir, los atributos de la relación no dependen unos de otros, dependen únicamente de la clave, esté formada por uno o más atributos. La dependencia $X \rightarrow Z$ es transitiva si existen las dependencias $X \rightarrow Y$, $Y \rightarrow Z$, siendo X , Y , atributos o conjuntos de atributos de una misma relación.

Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que dependen).

Teorema de la 3FN: sea una relación formada por los atributos A, B, C con clave primaria formada por el atributo A . Si se cumple que: $B \rightarrow C$, entonces la relación puede descomponerse en dos relaciones, $R1$ y $R2$, con los atributos respectivos: $R1(A, B)$ y $R2(B, C)$.

Caso práctico

12 Supongamos que tenemos una relación LIBROS en la que representamos los datos de las editoriales de los mismos, ver Tabla 2.11:

COD_LIBRO	TÍTULO	EDITORIAL	PAÍS
12345	DISEÑO DE BD RELACIONALES	RAMA	ESPAÑA
34562	INSTALACIÓN y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL	ESPAÑA
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA	ESPAÑA
34522	BASE DE DATOS OO	ADDISON	EEUU

Tabla 2.11. Relación LIBROS para transformar a 3FN.

Veamos las dependencias con respecto a la clave:

- TÍTULO y EDITORIAL dependen directamente del código del libro.
- EL PAÍS, aunque en parte también depende del libro, está más ligado a la EDITORIAL a la que pertenece el libro. Por esta última razón, la relación libros no está en 3FN, la solución se muestra en las Tablas 2.12 y 2.13:

COD_LIBRO	TÍTULO	EDITORIAL (FK)	EDITORIAL	PAÍS
12345	DISEÑO DE BD RELACIONALES	RAMA	RAMA	ESPAÑA
34562	INSTALACIÓN y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL	MCGRAW-HILL	ESPAÑA
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA	SANTILLANA	ESPAÑA
34522	BASE DE DATOS OO	ADDISON	ADDISON	EEUU

Tabla 2.12. Tabla LIBROS en 3FN.

Tabla 2.13. Tabla EDITORIAL en 3FN.



2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



Actividades propuestas

- 2 Dada la tabla 2.14 normalizar hasta la 3FN.

DNI	NOMBRE	APELLIDOS	DIRECCIÓN	C_POST	POBLACIÓN	PROVINCIA
413245-B	JUAN	RAMOS	C/Las cañas 59 C/Pilón 12	19005 45589	GUADALAJARA CALERUELA	GUADALAJARA TOLEDO
23456-J	PEDRO	PÉREZ	C/Vitoria 3 C/El altozano	28804 10392	ALCALÁ DE HENARES BERROCALEJO	MADRID CÁCERES
34561-B	MARÍA	RODRÍGUEZ	C/Sanz Vázquez 2	19004	GUADALAJARA	GUADALAJARA
222346-J	JUAN	CABELLO	C/El ensanche 3 C/Los abedules 10	28802 10300	ALCALÁ DE HENARES NAVALMORAL DE LA MATA	MADRID CÁCERES

Tabla 2.14. Transformar a 3FN.

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

Se define determinante en una relación a un atributo del cual depende funcionalmente de manera completa cualquier otro atributo de la relación. Una relación está en la Forma Normal de Boyce - Codd (FNBC) si, y sólo si, todo determinante de ella es una clave candidata.

La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si las hubiera. La BCFN es más fuerte que la 3FN, por lo tanto, toda relación en FNBC está en 3FN.

La violación de la FNBC es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación viola la FNBC si tiene dos o más claves candidatas compuestas que tienen al menos un atributo en común.

Teorema de Boyce - Codd: sea una relación R formada por los atributos A, B, C, D con claves candidatas compuestas (A, B) y (B, C) tal que: A C, entonces la relación puede descomponerse en cualquiera de las dos siguientes maneras: R1 (A, C) y R2 (B, C, D) o bien, R1 (A, C) y R2 (A, B, D).



Caso práctico

- 13 Supongamos que tenemos una relación EMPLEADOS en la que representamos los datos de los empleados de una fábrica, ver Tabla 2.15:

DNI	NÚM_SEG_SOC	NOMBRE	APELLIDOS	DEPARTAMENTO	PUESTO	SALARIO
413245-B	28-1234566	JUAN	RAMOS	COMPRAS	GERENTE	2.300
23456-J	28-2345686	PEDRO	PÉREZ	NÓMINAS	AUXILIAR	1.200
123123-C	19-458766	MARÍA	GIL	ALMACÉN	CONSERJE	1.530
1234556-B	45-223344	ANTONIO	SANZ	COMPRAS	GESTIÓN	2.200

Tabla 2.15. Relación EMPLEADOS para transformar a FNBC.

(Continúa)

2. El modelo de datos relacional

2.7 Normalización de esquemas relationales



(Continuación)

Observando los atributos de esta relación podemos ver que NUM_SEG_SOC y el grupo NOMBRE-APELLIDOS son claves candidatas (determinantes). Esta relación se transforma en dos tablas: una contendrá la clave junto con las claves candidatas (EMPLEADOS) y la otra la clave con el resto de campos (EMPLE_TRABAJO), ver Tablas 2.16 y 2.17:

DNI	NUM_SEG_SOC	NOMBRE	APELLIDOS
413245-B	28-1234566	JUAN	RAMOS
23456-J	28-2345686	PEDRO	PÉREZ
123123-C	19-458766	MARÍA	GIL
1234556-B	45-223344	ANTONIO	SANZ

Tabla 2.16. Relación EMPLEADOS e FNBC.

DNI (FK)	DEPARTAMENTO	PUESTO	SALARIO
413245-B	COMPRAS	GERENTE	2.300
23456-J	NÓMINAS	AUXILIAR	1.200
123123-C	ALMACÉN	CONSERJE	1.530
1234556-B	COMPRAS	GESTIÓN	2.200

Tabla 2.17. Relación EMPLE_TRABAJO en FNBC.

- Cuarta Forma Normal (4FN)

Una relación se encuentra en 4FN si, y sólo si, está en FNBC y no existen dependencias multivaluadas.

Teorema de Fagin: Dada la relación formada por los atributos X, Y, Z con las siguientes dependencias multivaluadas: $X \rightarrow \rightarrow Y$ y $X \rightarrow \rightarrow Z$, entonces la relación puede descomponerse en dos relaciones: R1 (X, Y) y R2 (X, Z).

Caso práctico

- 14 Consideramos la siguiente relación: GEOMETRÍA, con tres atributos con dependencias multivaluadas, los tres atributos forman la clave, ver Tabla 2.18:

FIGURA	COLOR	TAMAÑO
ESFERA	ROJO	GRANDE
ESFERA	VERDE	GRANDE
CUBO	BLANCO	GRANDE
CUBO	AZUL	GRANDE
PIRÁMIDE	BLANCO	MEDIANO
PIRÁMIDE	BLANCO	GRANDE
PIRÁMIDE	ROJO	GRANDE

Tabla 2.18. Relación GEOMETRÍA con dependencias multivaluadas.

(Continúa)



2. El modelo de datos relacional

2.7 Normalización de esquemas relationales

(Continuación)

En la Tabla 2.18, FIGURA determina valores múltiples de COLOR y TAMAÑO, pero COLOR y TAMAÑO son independientes entre sí. Así pues, las dependencias son $\text{FIGURA} \rightarrow \rightarrow \text{COLOR}$ y $\text{FIGURA} \rightarrow \rightarrow \text{TAMAÑO}$. Observamos que se repiten ESFERA-GRANDE, o PIRÁMIDE-BLANCO, o PIRÁMIDE-GRANDE. Para eliminar la redundancia de los datos se deben eliminar las dependencias de valores múltiples. Esto se logra construyendo dos tablas, donde cada una almacena datos para solamente uno de los atributos de valores múltiples.

Estas repeticiones hacen que la tabla no esté en 4FN. Para pasarla a 4FN, creamos las Tablas 2.19 y 2.20:

<u>FIGURA</u>	<u>COLOR</u>
ESFERA	ROJO
ESFERA	VERDE
CUBO	BLANCO
CUBO	AZUL
PIRÁMIDE	BLANCO
PIRÁMIDE	ROJO

Tabla 2.19. Tabla COLOR en 4FN.

<u>FIGURA</u>	<u>TAMAÑO</u>
ESFERA	GRANDE
CUBO	GRANDE
CUBO	MEDIANO
PIRÁMIDE	GRANDE

Tabla 2.20. Tabla TAMAÑOS en 4FN.



Actividades propuestas

- 3 Normaliza la Tabla 2.21. Suponemos que los estudiantes de una academia pueden inscribirse en varias especialidades y en varios cursos. Por ejemplo, el número de matrícula 12345 tiene especialidades en Sistemas y Telemática y toma los cursos de Natación y Danza.

<u>NUMATRICULA</u>	<u>ESPECIALIDAD</u>	<u>CURSO</u>
12345	SISTEMAS	NATACIÓN
12345	TELEMÁTICA	DANZA
34567	SISTEMAS	PIANO
34567	DESARROLLO	NATACIÓN
67891	TELEMÁTICA	AERÓBIC

Tabla 2.21. Tabla ACADEMIA.

- **Quinta Forma Normal (5FN)**

Una relación se encuentra en 5FN si, y sólo si, toda dependencia de reunión en la relación es una consecuencia de las claves candidatas. Esto es, la relación estará en 5FN si está en 4FN y no existen restricciones impuestas por el creador de la base de datos. La 5FN se refiere a dependencias que son extrañas. Tiene que ver con tablas que pueden dividirse en subtablas, pero que no pueden reconstruirse. Su valor práctico es ambiguo ya que conduce a una gran división de tablas.

2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



2.8 Dinámica del modelo relacional: álgebra relacional

Hasta ahora, hemos visto la estructura y restricciones del modelo relacional. Es lo que se llama la **componente estática o estática del modelo relacional**. Todo modelo de datos lleva asociado a su parte estática una dinámica que permite la transformación de un estado origen a un estado final de la base de datos. Esta transformación se realiza aplicando un conjunto de operadores mediante los cuales se podrán realizar operaciones de inserción, borrado, modificación y consulta de tuplas.

La dinámica del modelo relacional actúa sobre conjuntos de tuplas y se expresa mediante lenguajes de manipulación relacionales que asocian una sintaxis a las operaciones. En lo que al álgebra se refiere la dinámica la constituye una serie de operadores que aplicados a las relaciones dan como resultado nuevas relaciones.

Operaciones básicas sobre tablas

En este apartado, se muestran diferentes tipos de operaciones de consulta que se pueden realizar con las tablas. Estas operaciones están basadas en el álgebra relacional. Los operandos de cada operación los constituyen una o varias tablas, y el resultado es otra tabla. Esta tabla resultante puede ser, a su vez, operando en otra operación.

Existen dos tipos de operaciones sobre tablas: *básicas* y *derivadas*. Las operaciones básicas se dividen, de nuevo, en *unarias* y *binarias*. En las operaciones unarias se utiliza una tabla de entrada para obtener el resultado; en las binarias se emplean dos tablas.

Las **operaciones derivadas** son binarias que necesitan las operaciones básicas, y son la *intersección*, el *cociente* y la *combinación o join*. La Figura 2.18 resume los tipos de operaciones básicas y derivadas.

OPERACIONES BÁSICAS	Operaciones unarias	Selección Proyección
	Operaciones binarias	Unión Diferencia Producto cartesiano
OPERACIONES DERIVADAS	Intersección Cociente Combinación	

Figura 2.18. Operaciones básicas y derivadas sobre tablas.

- **Operaciones básicas unarias**

- **Selección.** Esta operación obtiene un subconjunto de filas de una tabla con todas sus columnas. Se pueden seleccionar determinadas filas incluyendo en la operación una condición. En ésta se pueden utilizar los operadores booleanos: Y (AND), O (OR) , NO (NOT) para expresar varios criterios. Se representa de la siguiente manera: $\sigma_{\text{condición}}(\text{Tabla})$.



2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



Caso práctico

- 15 A partir de la tabla EMPLEADOS mostrada en la Figura 2.19, seleccionamos aquellas filas cuyo departamento es el 20.

Nº. EMPLE	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7499	ARROYO	2.080	390	30	7698
7521	SALA	1.625	650	30	7698
7566	JIMÉNEZ	3.867		20	7839
7654	MARTÍN	1.625	1820	30	7698
7698	NEGRO	3.705		30	7839
7782	CEREZO	3.185		10	7839
7788	GIL	3.900		20	7566
7839	REY	6.500		10	
7876	ALONSO	1.430		20	7788

Figura 2.19. Tabla EMPLEADOS con 10 filas.

Representamos la selección de la siguiente manera: $\sigma_{\text{NºDEPART}=20}(\text{EMPLEADOS})$. La tabla resultante se muestra en la Figura 2.20.

Nº. EMPLE	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7566	JIMÉNEZ	3.867		20	7839
7788	GIL	3.900		20	7566
7876	ALONSO	1.430		20	7788

Figura 2.20. Empleados del departamento 20.



Actividades propuestas

- 4 A partir de la tabla EMPLEADOS mostrada en la Figura 2.19, selecciona aquellas filas cuyo JEFE se corresponde con el número 7839 y el departamento es el 30.

- **Proyección.** Esta operación da como resultado una nueva tabla a partir de otra con el subconjunto de columnas indicado. Las filas duplicadas sólo aparecerán una vez. La proyección se representa de la siguiente manera: $\pi_{\text{col}_1, \text{col}_2, \dots}(\text{Tabla})$.

2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



Caso práctico



- 16 Proyectamos la tabla EMPLEADOS según las columnas APELLIDO y SALARIO. Se representa así: $\Pi_{\text{APELLIDO}, \text{SALARIO}}(\text{EMPLEADOS})$. La tabla resultante es la de la Figura 2.21.

APELLIDO	SALARIO
SÁNCHEZ	1.040
ARROYO	2.080
SALA	1.625
JIMÉNEZ	3.867
MARTÍN	1.625
NEGRO	3.705
CEREZO	3.185
GIL	3.900
REY	6.500
ALONSO	1.430

Figura 2.21. Proyección de la tabla EMPLEADOS según las columnas APELLIDO y SALARIO.

Obtenemos las columnas APELLIDO y SALARIO de aquellas filas de la tabla EMPLEADOS cuyo departamento es 20. Se representa así: $\Pi_{\text{APELLIDO}, \text{SALARIO}}(\sigma_{\text{NºDEPART}=20}(\text{EMPLEADOS}))$. La tabla resultante es la de la Figura 2.22.

APELLIDO	SALARIO
SÁNCHEZ	1.040
JIMÉNEZ	3.867
GIL	3.900
ALONSO	1.430

Figura 2.22. APELLIDO y SALARIO de los empleados del departamento 20.

Actividades propuestas



- 5 A partir de la tabla EMPLEADOS mostrada en la Figura 2.19, obtén el APELLIDO y SALARIO de aquellos empleados cuyo JEFE se corresponde con el número 7839 y el departamento sea el 30.

• Operaciones básicas binarias

- **Unión.** Dos tablas se pueden unir si tienen el mismo número de columnas y dominios compatibles. El resultado de la unión es otra tabla con las filas de ambas tablas. Las filas repetidas aparecen una sola vez. La unión de dos tablas se representa de la siguiente manera: Tabla1 \cup Tabla2.



2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



Caso práctico

- 17 La unión de las tablas EMPLE1 y EMPLE2 se muestra en la Figura 2.23.

EMPLE1		EMPLE2		EMPLE1 \cup EMPLE2	
N.º emple	Nombre	N.º emple	Nombre	N.º emple	Nombre
1001	Rosa	2001	Pilar	1001	Rosa
1005	Fernando	2010	Octavio	1005	Fernando

Figura 2.23. Unión de las tablas EMPLE1 y EMPLE2.

- **Diferencia.** La diferencia entre dos tablas sólo es posible si tienen el mismo número de columnas y dominios compatibles. El resultado es otra tabla con las filas pertenecientes a una de las tablas y no pertenecientes a la otra tabla. Se representa así: Tabla – Tabla2.



Caso práctico

- 18 La Figura 2.24 refleja la diferencia de las tablas EMPLE1 y EMPLE2.

EMPLE2 – EMPLE1		EMPLE1 – EMPLE2	
Nº. emple	Nombre	Nº. emple	Nombre
1001	Rosa	2001	Pilar

Figura 2.24. Diferencia de las tablas EMPLE1 y EMPLE2.



Actividades propuestas

- 6 A partir de las tabla AUTORES (Nombre, Editorial) y EDITORES (Nombre, Editorial) inventa un contenido para sus filas y columnas, y representa las operaciones de la unión y de la diferencia.

2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



- **Producto cartesiano.** Se puede realizar entre dos tablas que tengan diferente número de columnas. El resultado es otra tabla que contendrá la suma de columnas de ambas tablas y el conjunto formado por todas las filas de ambas tablas. No pueden existir columnas con el mismo nombre. El producto se representa de este modo: Tabla1 X Tabla2 .

Caso práctico

19 En la Figura 2.25 se muestra el producto cartesiano de las tablas VENTAS y ARTÍCULOS.

VENTAS			ARTICULOS			
Codi	Fecha	Cantidad	Código	Denominación	Existencias	PVP
5100	18/11/05	100	5100	Patatas	500	3,2
5200	19/11/05	120	5200	Cebollas	250	4,1
5100	19/11/05	45				

VENTAS X ARTICULOS						
Codi	Fecha	Cantidad	Código	Denominación	Existencias	PVP
5100	18/11/05	100	5100	Patatas	500	3,2
5100	18/11/05	100	5200	Cebollas	250	4,1
5200	19/11/05	120	5100	Patatas	500	3,2
5200	19/11/05	120	5200	Cebollas	250	4,1
5100	19/11/05	45	5100	Patatas	500	3,2
5100	19/11/05	45	5200	Cebollas	250	4,1

Figura 2.25. Producto cartesiano de las tablas VENTAS y ARTICULOS.

Actividades propuestas

7 A partir de las tablas EMPLEADOS (NºEmple, Apellido, Salario, Comision) y DEPARTAMENTOS (NºDepart, Nombre, Localidad) inventa un contenido para sus filas y columnas y representa la operación del producto cartesiano.

• Operaciones derivadas

- **Intersección.** Es una operación derivada de la diferencia. La intersección de dos tablas es otra tabla formada por las filas que aparecen en ambas tablas y las columnas de una de las tablas. Al igual que en la diferencia, las tablas han de tener el mismo número de columnas y dominios compatibles. La operación de la diferencia es la siguiente: Tabla1 - (Tabla1 - Tabla2). La intersección se representa de la siguiente manera: Tabla1 \cap Tabla2. La Figura 2.26 muestra la intersección de las tablas EMPLE1 y EMPLE2.



2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional

EMPLE1 \cap EMPLE2

Nº. emple	Nombre
1005	Fernando

Figura 2.26. Intersección de las tablas EMPLE1 y EMPLE2.

- **Cociente.** El cociente se realiza entre dos tablas, Tabla1 y Tabla2, con diferentes columnas que cumplen las siguientes condiciones: Tabla1 debe tener columnas de Tabla2 y el número de columnas ha de ser mayor que el de Tabla2. Tabla2 debe tener, al menos, una fila. Se puede expresar en función de la proyección, del producto cartesiano y de la diferencia de la siguiente forma:

$$\text{Tabla1 : Tabla2} = \Pi_C(\text{Tabla1}) - \Pi_C(\text{Tabla2} \times \Pi_C(\text{Tabla1}) - \text{Tabla1})$$

Donde C representa el conjunto de atributos de la Tabla1 menos los atributos de la Tabla2.



Caso práctico

- 20 En la Figura 2.27 se refleja el cociente de las tablas COCHES y COLORES. La expresión es la siguiente:

COCHES	
MODELO	COLOR
Seat	Rojo
Renault	Verde
Ford	Azul
Ford	Negro
Seat	Azul

COLORES	
COLOR	
Rojo	
Azul	

COCHES : COLORES	
MODELO	
Seat	

$$\text{COCHES : COLORES} = \Pi_{\text{MODELO}}(\text{COCHES}) - \Pi_{\text{MODELO}}(\text{COLORES} \times \Pi_{\text{MODELO}}(\text{COCHES}) - \text{COCHES})$$

MODELO
Seat
Renault
Ford
Ford
Seat

MODELO	COLOR
Seat	Rojo
Seat	Azul
Renault	Rojo
Renault	Azul
Ford	Rojo
Ford	Azul

MODELO	COLOR
Renault	Rojo
Renault	Azul
Ford	Rojo

MODELO
Renault
Ford

$$^*(\text{COCHES} \times \Pi_{\text{MODELO}}(\text{COLORES}) - \text{COCHES}) \\ \text{se eliminan filas duplicadas}$$

Figura 2.27. División de las tablas COCHES y COLORES.

Este operador es útil para simplificar consultas del tipo de la presentada en la figura anterior donde se desea obtener todos los modelos de coches para los que hay todos los colores, contenidos en la tabla COLORES.

2. El modelo de datos relacional

2.8 Dinámica del modelo relacional: álgebra relacional



Actividades propuestas



- 8 A partir de las tablas AUTORES(Nombre, Provincia, Editorial) y EDITORIALES(Editorial) inventa un contenido para sus filas y columnas y representar la operación de División (AUTORES:EDITORIALES). ¿Qué representa el resultado final?

- **Combinación o join.** Con esta operación se obtiene el producto cartesiano de dos tablas cuyas filas cumplan una condición determinada. La combinación de dos tablas con respecto a una cierta condición de combinación es otra tabla constituida por pares de filas de ambas tablas concatenadas tales que en cada par las correspondientes filas satisfacen la condición. Se representa del siguiente modo: $(\text{Tabla1} * \text{Tabla2})_{\text{condición}}$.

Caso práctico



- 21 La Figura 2.28 muestra la combinación de las tablas VENTAS y ARTÍCULOS vistas en el Caso práctico 19. La combinación de ambas tablas da lugar a una nueva tabla en la que las filas de cada venta contienen los datos del artículo correspondiente. El criterio de combinación es la igualdad en el código de artículo de ambas tablas. Los códigos de artículo están representados por las columnas CODI y CÓDIGO.

(CUENTAS X ARTÍCULOS) CODI=CÓDIGO

Codi	Fecha	Cantidad	Denominación	Existencias	PVP
5100	18/11/05	100	Patatas	500	3,2
5200	19/11/05	120	Cebollas	25	4,1
5100	19/11/05	45	Patatas	500	3,2

Figura 2.28. Combinación de las tablas VENTAS y ARTÍCULOS.

Actividades propuestas



- 9 A partir de las tablas EMPLEADOS (Nº.Emple, Apellido, Salario, Comisión, Nº.Depart, Jefe) y DEPARTAMENTOS(Nº.Departamento, NombreDepartamento, Localidad) inventa un contenido para sus filas y columnas y representa la operación de Combinación. Obtén después una expresión donde se obtengan sólo las columnas: Nº.Emple, Apellido, Salario y NombreDepartamento.



2. El modelo de datos relacional

Conceptos básicos

Conceptos básicos



La Figura 2.29 muestra un esquema básico de la estructura del modelo relacional.

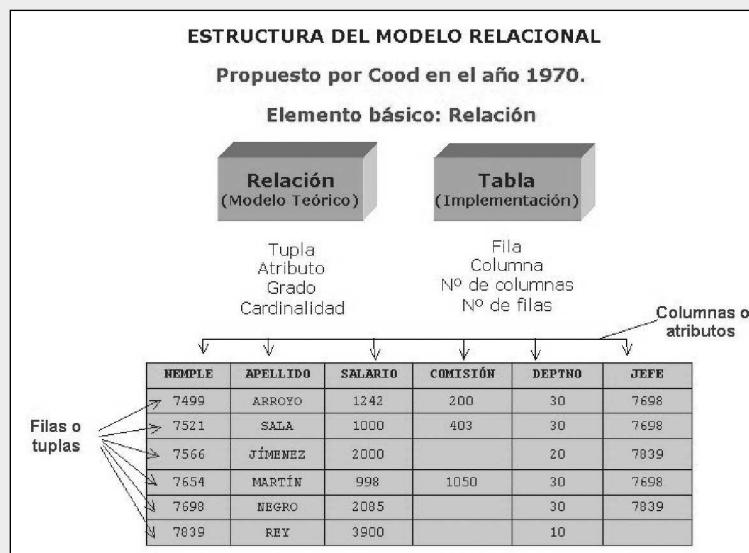


Figura 2.29. Estructura del Modelo relacional.

En la Figura 2.30 se muestra un esquema básico de las operaciones básicas sobre tablas basadas en el álgebra relacional.

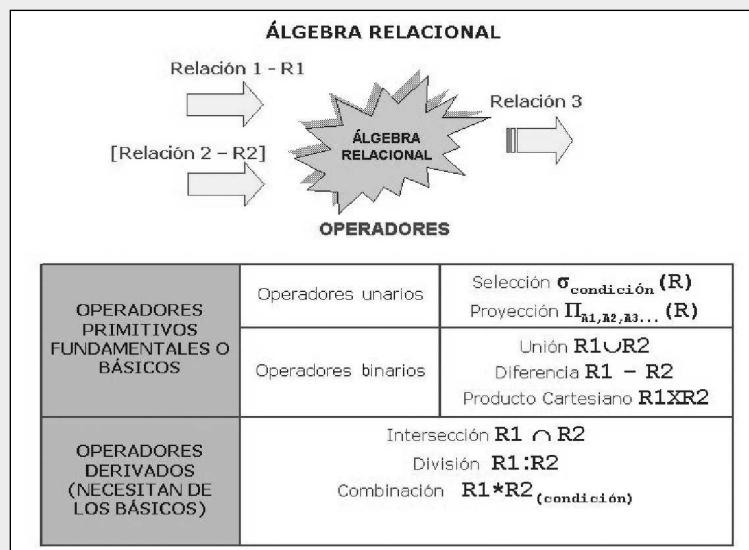


Figura 2.30. Operaciones básicas en álgebra relacional.

2. El modelo de datos relacional

Actividades complementarias



Actividades complementarias



1 Se desea informatizar la gestión de los proyectos del departamento de química de una universidad siguiendo las siguientes especificaciones:

- Al departamento llegan una serie de clientes que quieren realizar proyectos. Generalmente los clientes son empresas que realizan contratos con el grupo de investigación del departamento. Un cliente puede realizar varios proyectos.
- Un proyecto es de un cliente. Cada proyecto tiene asignada una cuantía de dinero que se utilizará para pagar los gastos del proyecto. De esta cuantía se saca el dinero para realizar los pagos a los colaboradores. También nos interesa saber de los proyectos el nombre, la fecha de comienzo, la de fin, entre otros.
- De cada proyecto se realizan muchos pagos para pagar a los colaboradores.
- De los pagos nos interesa saber el concepto, la cantidad, el IVA aplicado y la fecha del pago.
- Existen varios tipos de pagos (por ejemplo, Nómina, Representación, Material, etcétera). Un pago es de un tipo de pago, y a un tipo de pago pueden pertenecer muchos pagos.
- Existen una serie de colaboradores que son personas o entidades que van a recibir el dinero de los pagos en concepto de una tarea realizada o la compra de material. Un pago sólo puede ser para un colaborador. Éste a su vez puede recibir muchos pagos.
- De los colaboradores nos interesa saber: Nombre, NIF, Domicilio, Teléfono, Retención, Banco, Nº. Cuenta.

Realiza el diagrama E-R que cumpla las especificaciones y pásalo al modelo de datos relacional.

2 Dada la tabla 2.22, transformarla a 3FN.

COD_EMPL	NOMBRE	COD_DEP	NOMBRE_DEP	AÑOS_DEP
1	Juan	6	Contabilidad	6
2	Pedro	3	Sistemas	3
2	Pedro	6	Contabilidad	5
3	Sonia	2	I+D	1
4	Verónica	3	Sistemas	10
4	Verónica	6	Contabilidad	2

Tabla 2.22. Tabla ejercicio 2.

3 A partir de las siguientes tablas:

AGENDA (Nombre, Edad, CódigoProvincia, Telef)
PROVINCIAS (Código, Nombreprov)

- Escribe la columna o conjunto de columnas que pueden ser claves primarias y ajena.
- Escribe un enunciado para las siguientes expresiones:
 - $\sigma_{\text{Edad}>37}(\text{AGENDA})$
 - $\pi_{\text{Nombre}, \text{Edad}}(\text{AGENDA})$
 - $(\text{AGENDA} * \text{PROVINCIAS})$ CódigoProvincia = Código
 - $\pi_{\text{Nombre}, \text{CodigoProvincia}, \text{Telef}, \text{NombreProv}}((\text{AGENDA} * \text{PROVINCIAS})_{\text{CodigoProvincia} = \text{Código}})$
 - $\pi_{\text{Nombre}, \text{CodigoProvincia}, \text{Telef}, \text{NombreProv}}(\sigma_{\text{Edad}>37} ((\text{AGENDA} * \text{PROVINCIAS})_{\text{CodigoProvincia} = \text{Código}}))$

4 A partir de las siguientes tablas:

ALUMNOS(DNI, Nombre, Dirección, Telef, Curso)

ASIGNATURAS(Códigoasig, Nombreasig)

NOTAS(DNINo, Códigoasignatura, Nota)

- a) Define las claves primarias y ajena.
- b) Obtén expresiones relacionales para:
 - los alumnos de primer curso (Curso=1).
 - los alumnos de primer curso, sólo las columnas DNI y Nombre.
 - el Nombre de los alumnos, el Código de asignatura y la Nota de los alumnos de primer curso.
 - los DNI de los alumnos cuyo nombre de asignatura es INFORMÁTICA.