

# Módulo 2. Base de datos y modelo relacional

## Introducción

En el módulo 2, conoceremos lo que es un modelo de datos, y aprenderemos sobre los componentes que posee un sistema administrador de base de datos (DBMS, por sus siglas en inglés). Luego, ampliaremos sobre modelos de datos existentes como el de red, el jerárquico y, finalmente, el modelo relacional. Además del modelo relacional, analizaremos qué es un diagrama de entidad-relación, cuáles son las reglas de Codd, y qué es una dependencia funcional. Por último, profundizaremos en las diferentes formas normales y la normalización de base de datos, proceso que nos servirá para mejorar la calidad de nuestras bases de datos.

---

### Etapas 1: Creemos las entidades, atributos y relaciones de nuestra base de datos

## Video de inmersión

## Unidad 1. Sistemas de bases de datos

### Tema 1. Modelo de datos y sistemas de base de datos

De acuerdo con lo que establece Millán (2012):

Los modelos de datos están integrados por una serie de conceptos para describir los datos, sus relaciones y restricciones, y son útiles para representar, de manera abstracta, el mundo real. Su propósito es, además de facilitar la descripción de los datos y sus relaciones, permitir la representación de los datos y hacerlos comprensibles. Por esta razón, los modelos de datos facilitan el diseño de bases de datos. Para especificar la estructura y las restricciones, se usa un lenguaje de definición de datos (Data Definition Language, DDL, por sus siglas en inglés), y para especificar la manipulación de los datos se utiliza el lenguaje de manipulación de datos (Data Manipulation Language - DML, por sus siglas en inglés). Un DML ofrece mecanismos para recuperar datos de la base de datos vigente y actualizar datos. Algunas de las utilidades que tiene un modelo de datos son facilitar la especificación de los tipos y la forma en que los datos están organizados en una base de datos, y servir de base para desarrollar metodologías de diseño y lenguajes de alto nivel para consultar y manipular dichos datos. (p. 17)

Un modelo de datos debe tener lo siguiente:

Los autores identifican, al igual que en [Codd80], cuatro componentes lógicas que integran un modelo de base de datos: un conjunto de elementos atómicos y de relaciones entre estos elementos, denominado **espacio de datos**, una especificación de restricciones aplicadas a las relaciones en el espacio de datos, denominada **restricciones de definición de tipo**, un conjunto de operaciones para crear y destruir elementos y modificar las relaciones entre estos, denominada **operaciones de manipulación**, y un lenguaje de predicados para identificar de la base de datos elementos individuales por medio de sus propiedades lógicas. (Millán, 2012, p. 18)

---

### Sistema de gestión de base de datos

---

### Sistema de gestión de base de datos

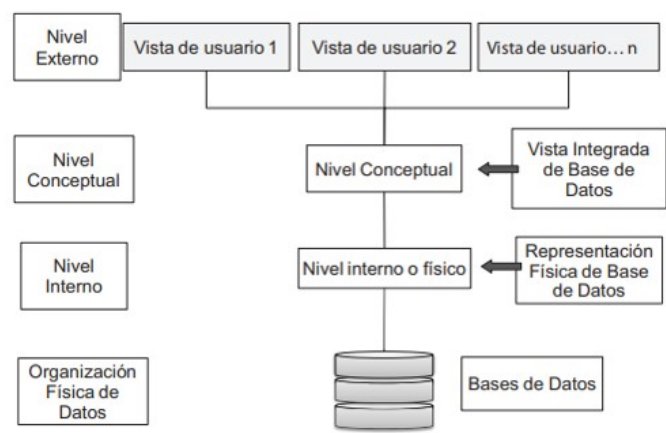
Siguiendo lo establecido por la autora:

Un sistema de gestión de bases de datos (SGBD) es una capa de *software* que necesitamos si deseamos crear, manipular y recuperar datos desde una base de datos. Como herramienta, sirve para realizar tareas generales, como estructurar, almacenar y controlar los datos, y provee interfaces de acceso a esta. Dentro de las tareas fundamentales que este sistema realiza, podemos mencionar los siguientes: mantenimiento de la integridad, seguridad de acceso, mecanismos de recuperación cuando surgen fallos tanto físicos como lógicos, control de concurrencia al acceso de los datos y eficiencia del sistema, que se evalúa, por lo general, según el tiempo que lleva responder preguntas de usuarios. Mediante el DDL y el DML, respectivamente, un usuario define una base de datos (tipos, estructura y restricciones) y puede recuperar, actualizar, insertar o borrar datos. Los usuarios no necesitan conocer detalles de almacenamiento de la base de datos, solo deben tener una vista abstracta de los datos. Por esta razón, la arquitectura de un SGBD, generalmente, se basa en la arquitectura de tres niveles (externo, conceptual e interno) ANSI-SPARC de la figura 1, tomada de [AA93]. Se trata de separar la forma en que los usuarios ven los datos de los detalles de almacenamiento físico de estos datos. Este principio de **independencia de datos** hace posible que el administrador de la base de

datos (BD) cambie la estructura física (nivel interno), sin que cambie la manera en la cual los diferentes usuarios ven los datos (nivel externo). El nivel interno describe la forma en la que los datos se almacenan en la base de datos (estructuras de datos, espacios de almacenamiento, índices, formato de registros). El nivel más bajo, el físico, trata con los mecanismos de almacenamiento físico que el sistema operativo utiliza (dispositivos físicos). Esto es lo que se conoce como «arquitectura funcional de una base de datos». (Millán, 2012, p. 19)

Veamos, entonces, los diferentes niveles que puede tener una base de datos, según su función.

Figura 1: Arquitectura funcional de una base de datos ANSI-SPARC



Fuente: [Imagen sin título sobre arquitectura]. (s.f.).

Rol del DBA

Como programadores, es común que interactuemos con el administrador de la base de datos. Por este motivo, es necesario conocer cuál es el rol que cumple, y veremos que, en algunos casos, podremos compartir tareas en común. Las tareas del administrador de la base de datos son las siguientes:

- Definir estructuras. Esto es la creación de la base de datos: tablas y sus relaciones.
- Definir estructuras físicas, es decir, qué archivos se van a crear para cada esquema.
- Especificar perfiles y seguridad: usuarios, roles y permisos a la base de datos.
- Implementar integridad: relaciones entre tablas.
- Definir métodos de respaldo y recuperación. Estas políticas se definen según la importancia y la sensibilidad de los datos.
- Monitorear el rendimiento, con el objetivo de optimizar consultas y el acceso a los datos para mejorar la experiencia del usuario final.

Sistema gerencial de una base de datos

Las siglas SGI o MIS en inglés, se utilizan para denominar un sistema gerencial de información. Un SGI es el resultado de la interacción colaborativa de varias partes, como, por ejemplo, personas, tecnologías y procedimientos. Habitualmente, para analizar la información, se alimentan de otros sistemas de la empresa u organización. Sus funciones gerenciales son planificación, dirección, control y organización, y todas ellas son necesarias para un buen desempeño de la organización. Los SGI tienen como función ayudar a proporcionar información que cumpla con las siguientes características:

≡ Calidad
La información que se proporcione debe ser fidedigna y no debe contener errores de procesamiento.
≡ Oportuna
La información debe estar disponible, en el momento preciso en el que se la necesita.

≡ Cantidad
Su presentación se debe orientar al usuario, y debe mostrar mayor nivel de detalle a un operario y menor detalle a un gerente, para que cada nivel pueda tomar decisiones oportunas.
≡ Relevante
La información que se brinde se debe relacionar con sus tareas y responsabilidades.

Componentes de un DBMS

Detallamos, a continuación, los componentes de un DBMS y su definición.

Tabla 1: componentes del DBMS

Componente	Definición
Compilador DDL	Procesa las definiciones de los esquemas y almacena las descripciones de los esquemas en el catálogo.
Compilador de consultas	Analiza las consultas sintácticamente, los nombres de elementos y objetos. Traduce a formato interno.
Optimizador de consultas	Reconfigura, reordena, elimina redundancias y usa algoritmos e índices para mejorar la ejecución de la consulta. Genera código ejecutable.
Precompilador DML	Extrae los comandos DML y los envía al compilador DML. Envía el resto del programa al compilador del lenguaje <i>host</i> .
Compilador DML	Compila y genera código objeto que le pasa al procesador de DB <i>runtime</i> .
Compilador de lenguaje <i>host</i>	Compila el programa y lo prepara para enlazar la transacción.
Procesador de BD <i>runtime</i>	Recibe comandos y los ejecuta, accede al diccionario de datos y se comunica con el sistema operativo (SO) para entrada/salida (E/S). Administra el <i>buffer</i> en la memoria de procesamiento, que se comparte o no con SO.
Subsistema de control de concurrencia	Subfunción del procesador de BD para colaborar con el control de concurrencia.
Administración de datos almacenados	Funciona con el procesador de BD para poder comunicarse con el SO y realizar E/S.

Fuente: elaboración propia

Para comprender mejor cada uno de estos componentes y sus relaciones, podemos observar, en la siguiente gráfica, cómo es su interacción:

Figura 2: Componentes del DBMS



En todo sistema de base de datos, es vital contar con una copia de los datos originales, para poder recuperarlos en caso de que ocurra una pérdida parcial o total de la información.

La contrapartida del proceso de copia de seguridad es el proceso de restauración de los datos.

Las copias de seguridad pueden almacenarse en diferentes dispositivos, cada uno de los cuales posee ventajas y desventajas. Al momento de seleccionar uno, hay que tener en cuenta diferentes aspectos, como, por ejemplo, facilidad de traslado, duplicidad de información, seguridad de los datos.

Hay diferentes estrategias de *backups* o respaldos, pero, a la hora de seleccionar una, debemos tener en cuenta la capacidad del almacenamiento disponible, si pensamos en duplicar datos. Lo usual es hacer una copia inicial y luego hacer copia de las modificaciones que van ocurriendo. Este tipo de resguardo se denomina «incremental», ya que cada copia **no** posee la totalidad de los datos, sino que contiene la diferencia de aquellos datos modificados.

Las copias de seguridad garantizan la integridad y la disponibilidad de una base de datos.

Tipos de respaldo

≡ Depósito del sistema de archivos

Es un tipo de copia de seguridad rápida. Como característica negativa, exige que la base de datos no se esté usando en el momento de la copia, lo cual, muchas veces, no es factible en los sistemas que poseen una alta disponibilidad para el usuario.

≡ Control de cambios

Este tipo de respaldo consiste en copiar los archivos que solo han sido modificados. Varios sistemas de archivos cuentan con un bit de archivo para cada archivo, que nos permite identificar si el archivo fue modificado o no, luego de la copia anterior. Si ha sido modificado, se resguarda, en caso contrario, no. Otra forma de saber si el sistema de archivos ha sido modificado es mirando la fecha de creación o modificación del archivo.

≡ Incremental a nivel de bloque

Una vez que se hace la copia base o primera copia, una forma de hacer una copia de seguridad de los archivos es copiar los bloques físicos que han sufrido cambios. Este tipo de copia es más complejo.

≡ Incremental o diferencial binaria

Es similar a la anterior, pero toma las diferencias binarias entre el *backup* anterior y el actual. La ventaja de este tipo de *backup* es que, al trabajar a nivel de byte, ahorra espacio y no depende del sistema operativo.

≡ Versionado del sistema de archivos

Se basa en los cambios del archivo, y crea estos cambios accesibles al usuario. Este tipo de copia de seguridad está integrada al ambiente informático.

Las unidades de cambio que maneja el sistema de bloques se basan en **unidades de cambio relativamente grandes** (bloques de 8 ks, 4 ks, 1 k).

Copia de seguridad de datos en uso

Si al momento de ejecutarse el *backup* la base de datos está siendo usada, corremos el riesgo de que existan archivos abiertos y de que se esté trabajando sobre esos archivos. Si esto sucede, posiblemente, el contenido del disco no refleje exactamente lo que el usuario ve. Nos resulta raro que esto pueda ocurrir, pero es muy frecuente.

Sí debemos tener en cuenta que este tipo de copia de seguridad suele tardar más tiempo. A fin de copiar un archivo en uso, es muy importante que la copia de seguridad entera represente un único paso, lo cual es todo un desafío cuando se está copiando un archivo en continua modificación. En estos casos, el archivo de base de datos está bloqueado para evitar cambios, pero debemos implementar un método para asegurar que el *snapshot* original es preservado con tiempo de sobra para ser copiado, incluso cuando se mantengan los cambios.

La operación de recuperación de la información se realiza de manera inversa a la operación de *backup* o resguardo, y se realiza de acuerdo con la técnica de *backup* que se haya implementado.

**Tener copias de los datos originales no posee importancia.**

Verdadero

Falso

Es de vital importancia tener siempre copias de resguardo de los datos. Cuanto mayor sensibilidad y disponibilidad tengan los datos, más imprescindible se hace tener no solo las copias de los datos, sino también mecanismos para recuperar y restaurar dichos datos.

## Tema 2. Modelo jerárquico, de red y relacional

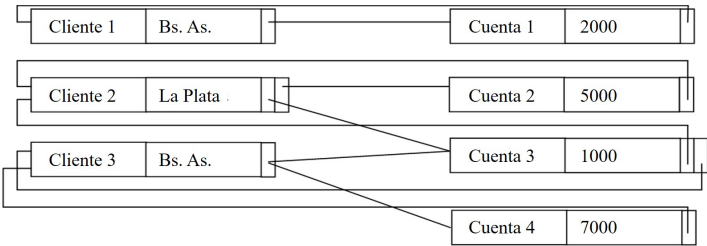
### Generación de estructuras relacionales

#### Modelo de red

Los datos son representados como colección de registros, de estructura arbitraria. Las relaciones entre estos son punteros o enlaces, y permiten relaciones de uno a muchos, y de muchos a muchos. Como desventaja, podemos mencionar que poseen una estructura compleja para problemas simples. Tienen algunas restricciones al momento de su aplicación, como, por ejemplo, no poder insertar el detalle de una factura (hijo), si no hay una cabecera de factura (padre). Las órdenes implican acciones físicas:

- find: Localiza un registro por condición y ubica el puntero.
- get: copia un registro señalado por el puntero actual al buffer.
- store: crea un registro con valores de datos ubicados en la memoria.
- modify: implica encontrarlo, subirlo a la memoria y modificarlo.
- erase: elimina registro, con previa búsqueda.
- connect: conecta un registro con un conjunto de registros.
- disconnect: desconecta un registro de un conjunto de registros.
- reconnect: mueve un registro de un conjunto a otro.

Figura 3: Modelo de red



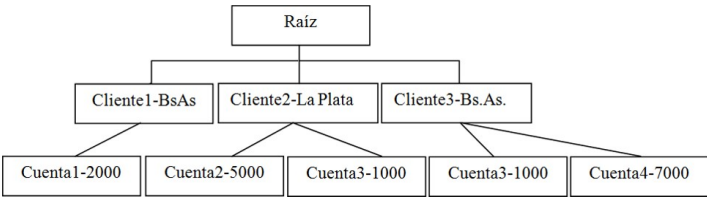
Modelo jerárquico

El modelo jerárquico es una aplicación particular del modelo de red. Los datos se representan como colección de registros, pero con estructura padre/hijo. Las relaciones también son punteros o enlaces, y permiten las relaciones de uno a uno y de uno a muchos. Como restricción, podemos mencionar que todo hijo tiene un solo padre, y que un padre puede tener uno o más hijos.

Las órdenes implican acciones físicas:

- get/where. Localiza y copia registro a memoria.
- insert: se localiza a su registro padre y luego inserta.
- replace: modifica uno o más campos de un registro.
- delete: elimina el registro actual de la base de datos.

Figura 4: Modelo jerárquico



Modelo relacional

Normalización

El modelo relacional tiene base en el concepto de **relación matemática**, más precisamente en la teoría de conjuntos:

Intuitivamente, las relaciones se asocian con **tablas** nombradas cuyas columnas representan **atributos** que también tienen asociado un nombre. Las filas de las tablas se denominan **tuplas**. Los **valores** que toman esas tuplas se extraen de conjuntos de constantes, llamados **dominios**. Todas las tablas constituyen la estructura de la base de datos que se representa en un esquema de base de datos (nivel intencional) y su contenido en una instancia de base de datos (nivel extensional). La propuesta inicial del modelo de datos relacional caracteriza las relaciones como la estructura fundamental para describir y organizar los datos y el álgebra relacional para manipularla.

El modelo relacional se define integrado a partir de tres elementos:

- un conjunto de relaciones que varía en el tiempo;
- reglas de inserción-actualización-eliminación;
- un álgebra relacional. (Millán, 2012, p. 25).

Para diseñar un modelo de datos relacional, se utiliza una serie de normas o reglas denominadas **formas normales**, que establecen el fundamento teórico de la **normalización**.

Características o propiedades de las relaciones:

- las tuplas no están ordenadas (conjunto matemático).
- Los atributos en las tuplas no están ordenados (conjunto matemático).
- Los valores en los atributos deben ser atómicos (relaciones normalizadas).
- No hay tuplas repetidas (hechos diferentes del mundo real).

---

## Tema 3. Modelado y diseño de una BD relacional

### Modelo relacional

Las bases de datos recopilan datos, que se pueden manipular con facilidad, y se pueden mostrar o presentar de diversas maneras. El proceso de construir una base de datos se denomina «diseño de base de datos».

La definición del problema es el proceso por el cual se determina la organización de una base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar.

Durante mucho tiempo, el diseño de bases de datos se consideró una tarea para expertos: más un arte que una ciencia. Sin embargo, se ha progresado mucho en el diseño de bases de datos, y este se considera ahora una disciplina estable, con métodos y técnicas propios.

La aceptación de las bases de datos en la industria y el gobierno, en el plano comercial, se ha incrementado. Además, existe una pluralidad de aplicaciones técnicas y científicas. Estos dos factores han provocado que el diseño de bases de datos tenga hoy un rol esencial en el empleo de los recursos de información, en la mayor parte de las organizaciones. El diseño de bases de datos ha pasado a constituir parte de la formación general de los informáticos, en el mismo nivel que la capacidad de construir algoritmos mediante un lenguaje de programación convencional (Costal Costa, 2017).

### Captura de requerimientos

Para definir correctamente el problema, lo primero que debemos hacer es realizar un diseño conceptual que parta de las especificaciones de los requisitos del usuario. El resultado será el esquema conceptual de la base de datos, que corresponderá a un modelo entidad-relación (E/R). Un esquema conceptual es una descripción de alto nivel de la estructura de la base de datos, independientemente del DBMS que se vaya a utilizar para manipularla.

Un modelo conceptual es un lenguaje que se utiliza para describir esquemas conceptuales. El objetivo del diseño conceptual es describir el contenido de los datos de la base de datos (DB), y no las estructuras de almacenamiento que se necesitarán para manejar esta información.

El modelo relacional representa un sistema de bases de datos en un nivel de abstracción un tanto alejado de los detalles de la máquina subyacente. De hecho, el modelo relacional puede considerarse un lenguaje de programación más bien abstracto, que se orienta, de manera específica, a las aplicaciones de bases de datos.

Podemos asemejar una relación a un archivo, una tupla a un registro y un atributo a un campo. Sin embargo, estos parecidos son aproximados en el mejor de los casos. Una relación no debe pensarse como un solo archivo, sino como un archivo disciplinado, gracias a lo cual se simplifican considerablemente las estructuras de datos con las que debe interactuar el usuario, que, asimismo, facilita los operadores necesarios para manejar dichas estructuras.

### Generación de estructuras relacionales

---

#### Clave primaria

---

El primer paso para la definición del problema, el diseño de una base de datos, es la producción del esquema conceptual. Normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información. Cada una de estas visiones suele corresponder a las diferentes áreas funcionales de la empresa, como, por ejemplo, producción, ventas, recursos humanos, etcétera. A los esquemas conceptuales correspondientes a cada vista de usuario se los denomina «esquemas conceptuales locales». Cada uno de estos esquemas se compone de entidades, relaciones, atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo. Las tareas que se deben realizar en el diseño conceptual son las siguientes:



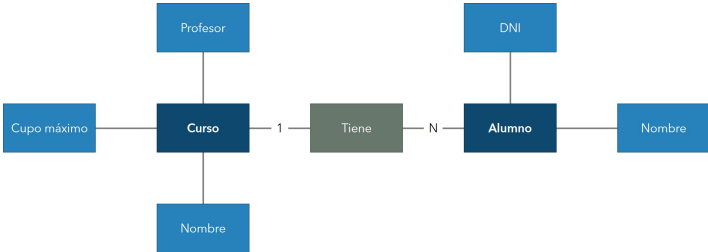
- a) identificar las entidades;
- b) identificar las relaciones;
- c) identificar los atributos y asociarlos a entidades y relaciones;
- d) determinar los dominios de los atributos;
- e) determinar los identificadores;
- f) determinar las jerarquías de generalización (si las hay);
- g) dibujar el diagrama entidad-relación;
- h) revisar el esquema conceptual local con el usuario.

El modelo entidad-relación (E/R) tiene base en una representación del mundo real en que los datos se describen como entidades, relaciones y atributos. El principal concepto del modelo E/R es la entidad, que es una «cosa» en el mundo real con existencia independiente. Una entidad puede ser un objeto físico (una persona, un auto, una casa o un empleado), o un objeto conceptual (una compañía, un puesto de trabajo o un curso universitario). En nuestro ejemplo de la sección anterior, podemos definir dos entidades: alumnos y cursos. Cada entidad tiene propiedades específicas, llamadas «atributos», que la describen. Por ejemplo, un curso tiene un nombre, un cupo máximo, un profesor.

Una entidad particular tiene un valor para cada uno de los atributos, y cada uno de ellos posee un dominio, que corresponde al tipo del atributo. Por ejemplo, matrícula tiene como dominio al conjunto de los enteros positivos, y nombre tiene como dominio al conjunto de caracteres. Para todo conjunto de valores de una entidad, debe existir un atributo o combinación de atributos que identifique a cada entidad de forma única. Este atributo o combinación de atributos se denomina «llave (primaria)». Por ejemplo, el número de matrícula es una buena llave para la entidad alumno, pero no así el nombre, porque pueden existir dos personas con el mismo nombre.

Una relación se puede definir como una asociación entre entidades. Por ejemplo, la entidad libro puede estar relacionada con la entidad persona por medio de la relación “está pedido”. La entidad curso puede estar relacionada con la entidad alumno mediante la relación «tiene». Siguiendo con la relación mencionada, “Curso-tiene-Alumno”, también surge la necesidad de indicar la cardinalidad de la relación. En este caso, tenemos que un (1) curso puede tener muchos (N) alumnos. Según lo que venimos detallando, podemos tener un modelo entidad-relación como el que se presenta en la figura 5.

Figura 5: Modelo entidad-relación Curso-Alumno



Fuente: elaboración propia.

Vemos, entonces, que las entidades se indican con un rectángulo, la relación con un rombo y enumerando las diferentes cardinalidades, y, por último, los atributos se indican con elipses o círculos. Siempre debemos indicar el nombre para cada elemento (entidad, atributo, relación).

De acuerdo con el modelo relacional, completá las siguientes oraciones:

Las ..... representan entidades del mundo real.

Tablas.

Tablets.

Columnas.

Las \_\_\_\_\_ representan los atributos de las entidades.

Entidades.
Columnas.
Pilares.

Los registros o filas de cada tabla se denominan \_\_\_\_\_.

Renglón.
Tuplas.
Quíntupla.

---

## Tema 4. Reglas de Codd y dependencia funcional

### Reglas de Codd

En 1970, el modo en el que se veían las bases de datos cambió por completo cuando E. F. Codd introdujo el modelo relacional. En aquellos momentos, el enfoque existente para la estructura de las bases de datos utilizaba punteros físicos (direcciones de disco), para relacionar registros de distintos ficheros. Si, por ejemplo, se quería relacionar un registro con otro, se debía añadir al registro un campo que contuviera la dirección en disco del registro. Este campo añadido, un puntero físico, siempre señalaría desde el registro al registro. Codd demostró que estas bases de datos limitaban, en gran medida, los tipos de operaciones que los usuarios podían realizar sobre los datos. Además, estas bases de datos eran muy vulnerables a los cambios en el entorno físico. Si se añadían los controladores de un nuevo disco al sistema, y los datos se movían de una localización física a otra, se requería una conversión de los ficheros de datos. Estos sistemas tenían base en el modelo de red y el modelo jerárquico, los dos modelos lógicos que constituyeron la primera generación de los DBMS.

El modelo relacional representa la segunda generación de los DBMS. En él, todos los datos están estructurados a nivel lógico (como tablas formadas por filas y columnas), aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica. Sin embargo, detrás de esa simple estructura, hay un fundamento teórico importante del que carecen los DBMS de la primera generación, lo que constituye otro punto a su favor.

Codd se dio cuenta de que existían bases de datos en el mercado que decían ser relacionales, pero vio que esas bases de datos solo guardaban la información en las tablas, que no estaban realmente normalizadas. Frente a esto, publicó las doce reglas que un verdadero sistema relacional debería tener. Ya sabemos que, en la práctica, es muy difícil que se apliquen todas las reglas. Mientras más reglas apliquen a un sistema, más «relacional» podrá considerarse.

#### Regla 1: La regla de la información

En un RDBMS, toda la información se representa, de manera explícita, de una única forma por los valores en una tabla. Todo aquello que no exista en una tabla, no existe del todo. Toda la información (nombres de tablas, nombres de vistas, nombres de columnas, datos de las columnas) se debe almacenar en tablas o dentro de las bases de datos. El diccionario de datos, en este sentido, está constituido por las tablas que presentan dicha información. En una base de datos relacional, la información se representa, de manera explícita, de una manera exacta en el nivel lógico, es decir, con valores en tablas. En este sentido, los metadatos (diccionario, catálogo) se representan exactamente

del mismo modo que los datos de usuario. Cabe destacar que es posible usar el mismo lenguaje (por ejemplo, SQL) para el acceso a los datos y los metadatos (regla 4).

#### **Regla 2: La regla del acceso garantizado**

Todos los ítems de datos deben ser, de manera lógica, accesibles, si se ejecuta una búsqueda en la que se combina el nombre de la tabla, su clave primaria, y el nombre de la columna. Es decir, a partir del nombre de una tabla, el valor de su clave primaria, y el nombre de la columna requerida, es posible encontrar uno y únicamente un valor. Por este motivo, es obligatorio definir las claves primarias para todas las tablas.

#### **Regla 3: Tratamiento sistemático de los valores nulos**

La información que falte o que no se pueda aplicar se puede representar a través de valores nulos. Un RDBMS (sistema gestor de bases de datos relaciones) debe tener la capacidad de soportar el uso de valores nulos en aquellas columnas en las que se desconozcan valores o en las que los valores no se puedan aplicar.

#### **Regla 4: La regla de la descripción de la base de datos**

La descripción de la base de datos se almacena de la misma forma en la que se almacenan los datos ordinarios, es decir, en tablas y columnas; además, los usuarios autorizados deben poder acceder a ella. La información de las tablas, las vistas, los permisos de acceso de usuarios autorizados, entre otros, se debe almacenar, de manera exacta, de la misma forma: en tablas. Al igual que todas las tablas, se debe poder acceder a ellas por medio de sentencias de SQL (o similar).

#### **Regla 5: La regla del sublenguaje integral**

Como mínimo, debe haber un lenguaje integral que soporte la definición de datos, su manipulación, la definición de vistas, las restricciones de integridad, y el control de autorizaciones y transacciones. Es decir, debe haber un lenguaje, con una sintaxis bien definida, que se pueda usar para administrar la base de datos por completo.

#### **Regla 6: La regla de la actualización de vistas**

Todas aquellas vistas que, en teoría, se pueden actualizar, se deben poder actualizar por el sistema mismo. La mayor parte de las RDBMS permite que se puedan actualizar vistas simples, pero deshabilitan los intentos de actualización de las vistas complejas.

#### **Regla 7: La regla de insertar y actualizar**

El manejo de una base de datos con operandos simples es válido tanto para la recuperación o la consulta de datos como para la inserción, actualización y el borrado de datos. Es decir, las cláusulas para leer, escribir, eliminar y agregar registros (SELECT, UPDATE, DELETE e INSERT en SQL) tienen que estar disponibles y deben ser operables, sin importar el tipo de relaciones y restricciones que se presente entre las tablas.

#### **Regla 8: La regla de independencia física**

El ingreso de usuarios a la base de datos —por medio de terminales o programas de aplicación— tiene que permanecer, de manera lógica, consistente, siempre que se quiera que haya modificaciones en los datos almacenados, o que se cambien los métodos de acceso a los datos. El comportamiento de los programas de aplicación y de la actividad de usuarios a través de terminales tendría que ser predecible, si se basa en la definición lógica de la base de datos, y no debería sufrir alteraciones, sin importar los cambios en la definición física de la base.

#### **Regla 9: La regla de independencia lógica**

Los programas de aplicación y las actividades de acceso por terminal tienen que quedar, de manera lógica, sin alteraciones, siempre que se quiera que se lleven a cabo modificaciones en las tablas de la base de datos (de acuerdo con los permisos que se asignen). La independencia lógica de los datos indica que los programas de aplicación y las actividades de terminales no deben depender de la estructura lógica; en este sentido, las modificaciones en la estructura lógica no deben alterar o cambiar estos programas de aplicación.

#### **Regla 10: La regla de la independencia de la integridad**

Todas las restricciones de integridad se deben definir con respecto a datos, y se deben almacenar con respecto a catálogo —no en el programa de aplicación.

##### **Las reglas de integridad**

La clave primaria **no** puede tener valores en blanco o nulos en ninguno de sus componentes (norma básica de integridad).

Para todos los valores de clave foránea, es fundamental que exista un valor de clave primaria concordante. La combinación de estas reglas asegura la integridad referencial.

#### **Regla 11: La regla de la distribución**

El sistema debe poder contar con un lenguaje de datos que soporte que la base de datos se distribuya, de manera física, en diferentes lugares, sin que ello afecte o altere los programas de aplicación. El soporte para bases de datos que se distribuyen implica que, una colección arbitraria de relaciones, de bases de datos, que corra en un conjunto de máquinas y sistemas operativos distintos, y que esté conectada por una variedad de redes, tenga la capacidad de funcionar como si estuviese disponible al igual que en una sola base de datos en una única máquina.

**Regla 12: Regla de la no subversión**

En caso de que el sistema cuente con lenguajes de nivel bajo, estos lenguajes, bajo ninguna circunstancia, se pueden usar para quebrantar la integridad de las reglas y restricciones que se expresan en un lenguaje de nivel alto (como, por ejemplo, SQL). Ciertos productos únicamente construyen una interfaz relacional para sus bases de datos no relaciones, lo cual posibilita la subversión (violación) de las restricciones de integridad. Esto no se debe permitir.

**Dependencia funcional**

Las formas normales son reglas que se deben utilizar en el diseño de una base de datos y el sustento conceptual de la normalización. Actualmente, hay cinco formas normales definidas que se mencionan por su orden, además de una especialización de la tercera forma normal que se denomina forma normal Boyce-Codd. Las formas normales tienen base en los siguientes conceptos:

- Dependencia funcional (DF) y dependencia funcional completa (DFC).
- Dependencia multivaluada (DMV).
- Dependencia de reunión.

Siendo a y b atributos de una misma relación T, se dice que b es funcionalmente dependiente de a si, para el mismo valor de a, tiene asociado un único valor de b, o, lo que es lo mismo, en todas las tuplas de T en las que el atributo a toma el mismo valor v1, el atributo b toma también un mismo valor v2.

La dependencia funcional se denota  $T.a \rightarrow T.b$  o bien simplemente  $a \rightarrow b$ .

Los atributos a y b pueden ser simples o compuestos (formados por la agregación de varios atributos). Los atributos funcionalmente dependientes pueden o no formar parte de la clave primaria de la tabla, de una clave candidata o de una clave foránea de otra tabla.

Claramente,  $a \rightarrow b$  no implica  $b \rightarrow a$ . Pueden repetirse los valores del atributo b para distintos valores de a (Universidad Autónoma del Estado de Hidalgo, 2017, <https://bit.ly/3PmPUXM>).

**Tabla 2: Datos de comprobantes**

Punto de venta y comprobante	Fecha	ID cliente	Nombre cliente	Total
01-100	1/3/17	01	María Paz	55
01-100	1/3/17	01	María Paz	55
02-101	21/3/17	02	Juan Casas	40
01-102	3/4/17	10	Ana Godoy	60
01-102	3/4/17	10	Ana Godoy	60
		A	B	

Fuente: elaboración propia.

Consideremos el ejemplo de una tabla que contiene datos de comprobantes. Podemos observar, claramente, que el atributo **nombre cliente** depende funcionalmente del **ID cliente**.

$a \rightarrow b$ .

**Dependencia funcional completa**

Dada una relación T, siendo x y z atributos compuestos pertenecientes a T, se dice que z depende funcionalmente de x de forma completa si z depende de x y no depende de ningún subconjunto de x. La dependencia funcional se representa como  $z \Rightarrow x$ . Si z es un atributo simple y  $z \rightarrow x$ , entonces la dependencia funcional es, con seguridad, completa.

### Dependencia multivaluada

En una relación T, con los atributos a, b y c, existe una dependencia multivaluada de b con respecto a si los posibles valores de b para un par de valores de a y c dependen únicamente del valor de a.

#### Características:

- Debemos observar que, para cada atributo de la relación, haya una evidente repetición de valores que nos haga pensar en redundancia de datos.
- La relación debe tener, por lo menos, tres atributos.
- Debe observarse que dos de los atributos sean independientes entre sí funcionalmente.
- El tercer atributo establece una dependencia funcional con los otros dos.

### Dependencia de reunión

Una relación R satisface la dependencia de reunión  $(DR)^*(A, B, Z)$  si y solo si R se puede construir con la reunión de sus proyecciones A, B, Z, siendo A, B, Z subconjuntos de atributos de R.

#### Características

- La cantidad de atributos de la entidad es tres o superior a tres.
- Todos los atributos de la relación componen la clave primaria.
- No debe haber dependencia funcional entre ninguno de sus atributos y tampoco dependencias multivaluadas no triviales.

La entidad se encuentra en 4FN.

## Unidad 2. Formas normales

### Normalización

Cuando se trabaja sobre la normalización de un modelo para resolver un problema, se está realizando un análisis de su lógica funcional. Esta acción de entender un problema permite identificar datos con características y propiedades que dan origen y establecen los atributos. Muchos de estos están ligados entre sí con determinada dependencia de funcionamiento.

Los atributos (datos) responden a una mecánica interna de funcionalidad que podríamos definir como procesos internos. Estos procesos dan el estado de asociación y dependencia de los atributos, y generan cierta definición de cómo se asocian estos entre sí. Las colecciones de atributos asociados semánticamente con un fin común permiten individualizar las entidades que conforman la estructura lógica. La aplicación de las formas normales permite que el entendimiento que se está realizando del problema se represente en un modelo normalizado de solución.

### Terminología

- Entidad.
- Atributo.
- Instancia.
- Dominio.

### Entidad

Se denomina entidad al contenedor de una colección de atributos que están fuertemente relacionados entre sí por una interpretación semántica, que determina a la entidad misma. Por ejemplo, si estás observando un problema que involucra personas, terminarás infiriendo

que debés asociar todas las características o los atributos de las personas en una entidad que las represente.

Una entidad comparte los mismos principios de definición que una relación, pero solo en el diseño de la estructura, sin hacer hincapié en sus características físicas y de almacenamiento.

### Atributo

Se denomina atributo a cada componente de una entidad que califica y establece una propiedad dentro de ella. En una entidad, los atributos tienen un nombre único y no poseen un orden específico.

### Instancia

Se denomina instancia a una ocurrencia de valores o al conjunto de valores que conforman un mismo grupo. Una instancia dentro de una entidad encuentra correspondencia con una tupla dentro de la relación. Las instancias son combinaciones ordenadas de valores irrepetibles. Una instancia de valores tiene una única posibilidad de existir dentro de la entidad. Las instancias no tienen orden.

### Dominio

El dominio, en el diseño lógico, es la caracterización conceptual del atributo, la definición de lo que queremos que represente ese atributo al que le estamos asignando el dominio. Esta definición está ligada al nombre propio del atributo y no al tipo de dato de este. Eso será motivo de análisis en una instancia posterior del modelo, en el diseño físico. Por este motivo, se recomienda establecer un nombre de atributo lo suficientemente representativo del dominio al que se quiere asignar, para que esto sea adecuado para su encasillamiento. Por ejemplo, si para una entidad personas establecemos un atributo de nombre fecha, su contenido puede ser amplio: fecha de nacimiento, de defunción, de ingreso o casamiento, o cualquier otra fecha que se le pueda atribuir a la persona; pero si el atributo se denomina fecha nacimiento, en él, solo se podrá almacenar este concepto.

### Formas normales

La dependencia funcional brinda la base conceptual para las siguientes formas normales:

- Primera forma normal [1FN o 1NF].
- Segunda forma normal [2FN o 2NF].
- Tercera forma normal [3FN o 3NF].
- Forma normal de Boyce-Codd [FNBC o NFBC].

La dependencia multivaluada solo actúa en la siguiente:

- Cuarta forma normal [4FN o 4NF].

---

### Cuarta forma normal

La dependencia de reunión solo actúa en la siguiente:

- Quinta forma normal [5FN o 5NF].

Si bien las formas normales son seis, los casos más comunes de normalización se suscitan en el rango que va de la primera forma normal

hasta Boyce-Codd.

## Tema 1. Primera forma normal

Una relación está en primera forma normal si y solo si todos los dominios simples subyacentes de los atributos poseen valores atómicos y monovalentes.

### Dominios simples de los atributos

Se refiere a que todos los atributos que componen una entidad tienen un único dominio y que, por ello, este es simple. Los atributos se encuentran reducidos conceptualmente a la mínima definición de dominio y, por este motivo, su expresión no resiste mayor división. Si el dominio es simple, como mencionamos, sus valores serán atómicos. La expresión **atómico** no significa que el contenido no pueda ser compuesto, como, por ejemplo, el nombre de una persona (María José), sino que el dominio **nombre** no está integrado a otro dominio, como el apellido.

### Atributos monovalentes

Hace referencia a que la información se representa una única vez dentro del modelo. Entonces, se deben arbitrar proyecciones (descomposiciones) de entidades suficientes para que la información sea reflejada una sola vez.

### Explicación desde la práctica

Observá la siguiente planilla de datos, que consta de un resumen de ventas de una librería que procesa todos sus datos desde una planilla de cálculos del tipo Excel. Tomaremos esta tabla y la procesaremos para que se transforme en N relaciones que cumplan con 1FN.

Tabla 3: Resumen de ventas

Pto. de venta y comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	Id producto	Nombre producto	Precio unitario	Cant.	Tota
01-100	01/03/17	01	María Paz	1	Tarjeta	1	Mapa	2,5	10	55
01-100	01/03/17	01	María Paz	1	Tarjeta	2	Hoja	1	30	55
02-101	21/03/17	02	Juan Casas	2	Contado	3	Sobre	4	10	40
01-102	03/04/17	10	Ana Godoy	3	Débito	1	Mapa	4,5	10	60
01-102	03/04/17	10	Ana Godoy	3	Débito	2	Hoja	1,5	10	60

Fuente: elaboración propia

### Dominios simples de los atributos

Primero, se deben observar los dominios de los atributos y detectar aquellos que no son dominios simples. Observá que el dominio de la columna **pto. (punto) de venta y comprobante** está compuesto por dos valores. Este dominio, evidentemente, no es simple. Los dos valores son el punto de venta y el número de comprobante respectivamente.

Tabla 4: Identificar los dominios que no son simples

Punto de venta y comprobante
01-100
01-100
02-101
01-102
01-102

Fuente: elaboración propia.

Tabla 5. Números identificados

Punto de venta	Número de comprobante
01	100
01	100
02	101
01	102
01	102

Fuente: elaboración propia.

Para lograr que la columna **Punto de venta y comprobante** posea dominio simple, se debe dividir en dos columnas: una para el punto de venta y otra para el número de comprobante.

Tabla 6: 1FN aplicada (dominio simple de los atributos)

Pto. de venta	N.º de comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	ID producto	Nombre producto	Precio/unidad	Cant.	Total
01	100	1/3/17	01	María Paz	1	Tarjeta	1	Mapa	2,5	10	25
01	100	1/3/17	01	María Paz	1	Tarjeta	2	Hoja	1	30	30
02	101	21/3/17	02	Juan Casas	2	Contad o	3	Sobre	4	10	40
01	102	3/4/17	10	Ana Godoy	3	Débito	1	Mapa	4,5	10	45
01	102	3/4/17	10	Ana Godoy	3	Débito	2	Hoja	1,5	10	15

Fuente: elaboración propia.

Atributos monovalentes

Ahora, deberíamos fijar nuestra atención en el concepto de **atributos monovalentes**. Todavía no se puede denominar **relación**, pues sigue sin respetar algunas restricciones necesarias para serlo. Otra forma de expresarlo sería evitar los grupos repetitivos, para que la información se exprese una sola vez.

Tabla 7: Identificando grupos repetitivos

Pto. de venta	N.º de comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	ID producto	Nombre producto	Precio/unidad	Cant.	Total
01	100	1/3/17	01	María Paz	1	Tarjeta	1	Mapa	2,5	10	55
01	100	1/3/17	01	María Paz	1	Tarjeta	2	Hoja	1	30	55
02	101	21/3/17	02	Juan Casas	2	Contado	3	Sobre	4	10	40
01	102	3/4/17	10	Ana Godoy	3	Débito	1	Mapa	4,5	10	60
01	102	3/4/17	10	Ana Godoy	3	Débito	2	Hoja	1,5	10	60

Fuente: elaboración propia.



Observá que, para los comprobantes 100 y 102, se repiten datos. Esto es porque en ambos casos hay dos productos detallados. Estos valores que se repiten no respetan la definición de monovalentes. Por lo tanto, se debe operar algún cambio para que los valores se presenten una sola vez. Las columnas que contienen valores repetidos son las que se resaltan con fondo amarillo. La acción que se debe aplicar es la separación de las columnas que contienen el grupo repetitivo en una tabla y las columnas que no contienen el grupo repetitivo en otra tabla.

Tabla 8: Comprobantes con datos repetitivos

Punto de venta	Número de comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	Total
01	100	1/3/17	01	María Paz	1	Tarjeta	55
01	100	1/3/17	01	María Paz	1	Tarjeta	55
02	101	21/3/17	02	Juan Casas	2	Contado	40
01	102	3/4/17	10	Ana Godoy	3	Débito	60
01	102	3/4/17	10	Ana Godoy	3	Débito	60

Fuente: elaboración propia.

Tabla 9: Detalle de comprobantes sin datos repetitivos

ID producto	Nombre producto	Precio unitario	Cantidad
1	Mapa	2,5	10
2	Hoja	1	30
3	Sobre	4	10
1	Mapa	4,5	10
2	Hoja	1,5	10

Fuente: elaboración propia.

Identificando claves primarias y foráneas

Ahora que tenemos dos tablas, deberíamos nombrarlas para poder individualizarlas. Cuando las mencionemos, simplemente diremos comprobantes o detalle comprobantes. Observá que tenemos dos tablas, pero no queda establecido en **detalle comprobantes**.

¿Qué detalle corresponde para cada comprobante?

También, siguen duplicadas las filas en la tabla comprobantes. La forma que tenemos para relacionar detalle comprobantes con comprobantes es estableciendo una clave foránea en detalle comprobantes que apunte a comprobantes. Para crear la FK en detalle de comprobantes, primero debemos tener declarada una clave primaria en comprobantes. Para ello, en la tabla o entidad comprobantes, no debe haber filas duplicadas. Eliminemos las filas duplicadas de comprobantes para poder declarar la PK.

Tabla 10: Comprobantes sin datos repetitivos y establecimiento de clave primaria

PK							
Punto de venta	Número de comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	Total
01	100	1/3/17	01	María Paz	1	Tarjeta	55
02	101	21/3/17	02	Juan Casas	2	Contado	40
01	102	3/4/17	10	Ana Godoy	3	Débito	60

Fuente: elaboración propia.

Aquí está la tabla de comprobantes sin filas duplicadas. Para establecer la PK, se debe elegir la mínima combinación irrepetible de valores que garanticen la unicidad. Los atributos de **punto de venta** y **número de comprobante** proporcionan esa condición de unicidad; entonces, serán la PK de **comprobantes**.

Al tener definida la PK de comprobantes, ya podemos establecer la FK en detalle comprobantes. La FK en detalle de comprobantes tiene que ser una estructura del mismo tipo que la PK de comprobantes.

Tabla 11: Detalle de comprobantes sin datos repetitivos y establecimiento de clave foránea

Punto de venta	Número de comprobante	ID producto	Nombre producto	Precio unitario	Cantidad
01	100	1	Mapa	2,5	10
01	100	2	Hoja	1	30
02	101	3	Sobre	4	10
01	102	1	Mapa	4,5	10
01	102	2	Hoja	1,5	10
FK					

Fuente: elaboración propia.

FK establecida, redundancia controlada. Ya tenemos establecida la relación entre ambas relaciones (dejaron de ser tablas). Lo único que falta es definir la PK de la tabla de detalle de comprobantes. Debemos establecer una combinación irrepetible de valores de atributos que permita identificar, de forma inequívoca, cada tupla de la relación detalle comprobantes. La combinación resaltada tiene características de unicidad. Esta será la PK de detalle comprobante.

Tabla 12: Detalle de comprobantes y establecimiento de clave primaria

PK					
Punto de venta	Número de comprobante	ID producto	Nombre producto	Precio unitario	Cantidad
01	100	1	Mapa	2,5	10
01	100	2	Hoja	1	30
02	101	3	Sobre	4	10
01	102	1	Mapa	4,5	10
01	102	2	Hoja	1,5	10
FK					

Fuente: elaboración propia.

Tabla 13: Definición del modelo en 1FN

Comprobantes			Detalle comprobantes		
PK	Punto de venta	PK	PK	Punto de venta	FK
	Comprobante			Comprobante	
	Fecha			ID producto	
	ID cliente			Nombre producto	
	Nombre cliente			Precio unitario	
	ID pago			Cantidad	
	Forma pago				

Fuente: elaboración propia.

Una relación está en segunda forma normal si y solo si primero está en primera forma normal y, además, todos los atributos no claves dependen por completo de la clave primaria. Es muy importante tener en claro que no se puede alcanzar la 2FN, si alguna relación del modelo no está en 1FN.

Cuando hacemos referencia a que todos los atributos no claves dependen por completo de la clave primaria, es importante entender que la expresión **atributos no claves** está expresada en su más amplio espectro. Entonces, los atributos tienen que no pertenecer ni a la clave primaria ni a ninguna clave candidata que pueda existir. Expresar atributos no claves equivale a decir atributos no primarios. Los **atributos no claves** deben depender por completo de la PK. Esto no se cumple cuando hay atributos no claves que dependen de una parte de la clave. Esta, entonces, indudablemente debe ser compuesta. Si la clave fuese simple, no hay posibilidad de encontrar esta situación.

Explicación desde la práctica

Debemos encontrar un atributo que dependa de una parte de la PK y no de la PK completa. Observá el atributo **nombre producto** de la relación **detalle comprobantes**. Este atributo tiene una fuerte dependencia con el **ID producto**, pues, al cambiar el código, necesariamente, debe cambiar el nombre. Para quebrar esa fuerte dependencia entre los atributos destacados, se debe descomponer la relación al sacar los atributos que dependen de **ID producto** a otra relación.

Tabla 14: Identificando atributos no claves que dependen de una parte de la clave primaria

PK					
Punto de venta	Número de comprobante	ID producto	Nombre producto	Precio unitario	Cantidad
01	100	1	Mapa	2,5	10
01	100	2	Hoja	1	30
02	101	3	Sobre	4	10
01	102	1	Mapa	4,5	10
01	102	2	Hoja	1,5	10
FK					

Fuente: elaboración propia.

Se deben despejar los atributos que dependen funcionalmente de **ID producto** a una nueva relación que las contenga. Para ello, la nueva relación (que llamaremos **productos**) debe tener los mismos atributos que se encuentran resaltados en la relación **detalle comprobantes**.

Ahora, debemos retirar los atributos de la relación **detalle comprobante** que causan que la 2FN no se cumpla y declarar la PK en la relación *productos*.

Observá que el precio sigue estando en **detalle comprobantes**. Esto es porque este atributo representa el valor en el momento en que se realizó la compra. Observá que la clave foránea que vincula **detalle comprobantes** con **productos** ya se encuentra declarada. Esto hace que ambas tablas mantengan una relación denominada **integridad referencial**.

Tabla 15: Relación detalle de comprobantes en 2FN

PK				
Punto de venta	Número de comprobante	ID producto	Precio unitario	Cantidad
01	100	1	2,5	10
01	100	2	1	30
02	101	3	4	10
01	102	1	4,5	10
01	102	2	1,5	10
FK		FK		

Fuente: elaboración propia.

Tabla 16: Relación productos en 2FN

PK		
ID producto	Nombre producto	Precio unitario
3	Sobre	4
1	Mapa	4,5
2	Hoja	1,5

Fuente: elaboración propia.

Definición de modelo en 2FN

Tabla 17: Definición del modelo en 2FN

Comprobantes			Detalle comprobantes			Productos		
PK	Punto de venta	PK	PK	Punto de venta	FK	PK	ID producto	Pk
	Comprobante			Comprobante			Nombre producto	
	Fecha			ID producto			Precio unitario	
	ID cliente			Nombre producto				
	Nombre cliente			Precio unitario				
	ID pago			Cantidad				
	Forma pago							

Fuente: elaboración propia.

Tema 3. Tercera forma normal y forma de Boyce-Codd

Una relación está en tercera forma normal si y solo si está en 2FN, y si ningún subconjunto de atributos no primarios tiene dependencia entre sí y como segunda medida dependen transitivamente de la clave primaria.

**Atributos no primarios:** es atributo no clave, que no forma parte de ninguna combinación de claves candidatas posibles.

La **dependencia transitiva** ocurre cuando dos atributos no primarios —que, por lo tanto, no forman parte de ningún tipo de clave— tienen una dependencia funcional entre sí más fuerte que la que se produce con la PK de la relación. Entonces, dependen entre sí uno del otro y, por ende, dependen transitivamente de la PK.

Explicación desde la práctica

Tenemos que observar dos o más atributos no claves (no primarios) que tengan una dependencia funcional entre sí más fuerte que la que posee con la PK de la relación. Esta situación solo se presenta en la relación **comprobantes**, entre los atributos **ID cliente** y **Nombre cliente**, por un lado, e **ID pago** y **Forma pago**, por otro.

Ambos atributos dependen uno del otro y no forman parte de ninguna clave.

Tabla 18: Relación comprobantes, identificando atributos no primarios con dependencia entre sí

PK							
Punto de venta	Número de comprobante	Fecha	ID cliente	Nombre cliente	ID pago	Forma pago	Total
01	100	1/3/17	01	María Paz	1	Tarjeta	55
02	101	21/3/17	02	Juan Casas	2	Contad o	40
01	102	3/4/17	10	Ana Godoy	3	Débito	60

Fuente: elaboración propia.

Se deben desagregar de esta relación los atributos que causan que no se alcance la 3FN. Para ello, se deben crear dos relaciones en las que se colocan los juegos de atributos que tiene dependencia funcional como sigue. Estos dos atributos tienen DF entre sí y luego dependen de forma transitiva de la PK. Declarará la PK de las nuevas relaciones para poder establecer las FK desde la relación que estamos analizando. Entre estos otros dos, también existe la misma situación. Ahora debe eliminar las columnas **nombre cliente** y **forma pago**, y establecer las FK para que la información quede integrada.

Tabla 19: Relación comprobantes, identificando no primarios con dependencia entre sí

PK					
Punto de venta	Número de comprobante	Fecha	ID cliente	ID pago	Total
01	100	01/03/17	01	1	55
02	101	21/03/17	02	2	40
01	102	03/04/17	10	3	60
			FK	FK	

Fuente: elaboración propia.

Tabla 20: Relación clientes

PK	
ID cliente	Nombre cliente
01	María Paz
02	Juan Casas
10	Ana Godoy

Fuente: elaboración propia.

Tabla 21: Relación forma de pagos

PK	
ID pago	Forma pago
1	Tarjeta
2	Contado
3	Débito

Fuente: elaboración propia.

Tabla 22: Definición del modelo en 3FN

Comprobantes			Detalle comprobantes			Productos		
PK	Punto de venta	PK	PK	Punto de venta	FK	PK	ID producto	pk
	Comprobante			Comprobante			Nombre producto	
	Fecha			ID producto			Precio unitario	

	ID cliente			Nombre producto		
	Nombre cliente			Precio unitario		
	ID pago			Cantidad		
	Forma pago					

Fuente: elaboración propia.

Tabla 23. Clientes y formas de pago

Clientes			Forma pagos		
PK	ID cliente	Pk	PK	ID pago	pk
	Nombre cliente			Forma pagos	

Fuente: elaboración propia.

Forma normal de Boyce-Codd

**Determinante:** uno o más atributos son determinantes cuando, de manera funcional, determinan a otro(s) atributo(s). En la dependencia funcional (A, B)-->C, (A, B) son los determinantes.

**Definición formal:** cuando un determinante es una llave candidata, una relación R está en FNBC. Una tabla se considera en esta forma si y solo si cada determinante o atributo es una llave candidata.

Tema 4. Cuarta forma normal

La 4NF se usa mucho en el diseño de base de datos, y garantiza que las dependencias multivaluadas independientes estén efectivamente correctas. La 4NF es el nivel siguiente de normalización después de la forma normal de Boyce-Codd.

Características

Una tabla se encuentra en 4NF si y solo si está en **tercera forma normal** y no presenta dependencias multivaluadas no triviales.

Explicación desde la práctica

Margaret S. Wu (1992) notó que, en la práctica, la normalización de base de datos, generalmente, solo alcanza hasta la 3FN. Esto se debe a la creencia de que las tablas violan la 4FN y, que, raramente las vamos a encontrar en aplicaciones organizacionales. Pese a esto, esta creencia puede no ser exacta.

Supongamos que poseemos una tabla con clientes.

Tabla 24: Relación clientes

PK			
ID cliente	Nombre cliente	Teléfono	Correo electrónico
01	María Paz	156145579	mpaz@gmail.com
02	Juan Casas	153467367	jasas@hotmail.com
10	Ana Godoy	155908786	anag@gmail.com

Fuente: elaboración propia.

¿Qué sucede si el cliente tiene más de un teléfono o dirección de correo electrónico? Podríamos tener una tabla como la siguiente.

Tabla 25: Relación clientes con más de un teléfono

PK			
ID cliente	Nombre cliente	Teléfono	Correo electrónico
01	María Paz	156145579 – 4895233	mpaz@gmail.com
02	Juan Casas	153467367	jasas@hotmail.com
10	Ana Godoy	155908786	anag@gmail.com –

Fuente: elaboración propia.

Como podemos observar, en este caso, se rompen las reglas de la normalización, desde la FN1 que indica que los campos deben ser atómicos. Una solución podría ser repetir las filas.

Tabla 26: Relación clientes con más de un teléfono

PK			
ID cliente	Nombre cliente	Teléfono	Correo electrónico
01	María Paz	156145579	mpaz@gmail.com
01	María Paz	4895233	mpaz@gmail.com
02	Juan Casas	153467367	jasas@hotmail.com
10	Ana Godoy	155908786	anag@gmail.com
10	Ana Godoy	155908786	ana.godoy@gmail.com

Fuente: elaboración propia.

Sin embargo, en este caso, si bien logramos la atomicidad de los campos, estamos rompiendo con otra FN, ya que estamos repitiendo los datos de ID cliente, nombre cliente y teléfono o correo electrónico, según corresponda. Lo que podemos observar es que tanto el teléfono como el correo son de dependencias multivaluadas con clientes, lo que nos lleva a una FN4 en la que se eliminan las dependencias multivaluadas.

Tabla 27: Relación clientes

PK			
ID cliente	Nombre cliente	Teléfono	Correo electrónico
01	María Paz	156145579	mpaz@gmail.com
01	María Paz	4895233	mpaz@gmail.com
02	Juan Casas	153467367	jasas@hotmail.com
10	Ana Godoy	155908786	anag@gmail.com
10	Ana Godoy	155908786	ana.godoy@gmail.com

Fuente: elaboración propia.

Tabla 28: Relación clientes y teléfono

PK	
ID cliente	Teléfono
01	156145579
01	4895233
02	153467367
10	155908786
10	155908786

Fuente: elaboración propia.

Tabla 29: Relación clientes y correo electrónico

PK	
ID cliente	Correo electrónico
01	mpaz@gmail.com
01	mpaz@gmail.com
02	jasas@hotmail.com

10	anag@gmail.com
10	ana.godoy@gmail.com

Fuente: elaboración propia.

### ¿Cuántas formas normales existen y qué características poseen?

Existen 3 formas normales y son superficiales, es decir, es necesario estar en la superficie.

Existen 6 formas normales y son exclusivas, es decir, es necesario no estar en la forma normal anterior, para considerarse de la siguiente.

Existen 4 formas normales y son inclusivas, es decir, la siguiente forma incluye a la anterior.

## Microactividad

### Caso: La distribuidora de películas

Una distribuidora nacional de películas desea mantener cierta información acerca de las películas que va a lanzar próximamente con el fin de promocionarlas adecuadamente.

Cada cierto tiempo se envían anuncios para insertar en los periódicos. Todos incluyen el título de la película, el nombre del actor o actores principales y el nombre del director (consideramos que hay un solo director por película). Además, para las películas que han recibido premios importantes Y el anuncio suele incluir un resumen de estos detallando el nombre del certamen, la ciudad donde se celebró y el tipo de premio recibido (mejor director, película, actor, etc.). Algunas veces se reseñan las mejores críticas, indicando el nombre del periódico o revista donde se publicó la crítica, la fecha de publicación y el autor que firma la crítica.

Otra parte importante de la promoción consiste en hacer de intermediarios entre los medios de comunicación y los participantes en la película, principalmente, el director, los productores y los actores.

Para concertar y planificar las citas es preciso mantener información personal acerca de los intervinientes en la película, como su nombre completo, dirección y teléfono de contacto. Además, con el fin de facilitar la elaboración del material de las entrevistas se proporciona a los medios de comunicación otros datos relacionados con la película: el aporte económico de cada productor, un resumen del argumento, un listado con el papel concreto que realiza cada actor en la película (hay que señalar que consideramos que un actor en una película hace un único papel), el tipo de papel (protagonista, secundario, etc.) y un resumen de los premios recibidos.

Es frecuente que determinados medios de comunicación nos soliciten información más personal sobre los actores como su fecha y lugar de nacimiento, su estado civil y sobre el sueldo que han cobrado por realizar la película. Las semanas previas al estreno se informa a los distintos medios de la fecha y el lugar del estreno de la película a nivel nacional. Anualmente la distribuidora realiza un estudio donde se incluye, entre otras cosas, un ranking de las películas más premiadas por nacionalidad.

### Consigna

Según la información dada por el cliente, armá el gráfico de entidad relación.

### Caso: Campeonato de fútbol

La organización de un campeonato de fútbol de selecciones nacionales decide implantar un sistema que les permita mantener información actualizada sobre diversos aspectos. Dos meses antes del comienzo deben conocerse los equipos que participarán (que habrán clasificado mediante unas liguillas previas de las cuáles nuestro sistema no guardará ninguna información).

De cada selección se debe conocer inicialmente, al menos, el país al que representa y el nombre del seleccionador. Y cada selección debe también facilitar una lista de jugadores inscritos con sus datos personales y el puesto en que juegan habitualmente (portero, defensa,



delantero centro, lateral izquierdo, etc.).

Un mes antes del comienzo se habrá concretado la planificación de los partidos estableciéndose la ciudad donde se jugará cada uno, la fecha y la hora de comienzo, los equipos que se enfrentarán, el nombre del colegiado principal que arbitrará el partido y el resto de colegiados que intervendrán en el mismo, así como al menos dos suplentes. Esta información se le remitirá oficialmente a cada selección, indicándole además qué equipos se enfrentarán en cada partido y cuál actuará de equipo local y cuál de visitante.

Por otra parte, a cada colegiado se le informará por carta de los partidos en que intervendrán y su función en el partido (árbitro principal, juez de línea o asistente). Una vez comenzado el mundial, diariamente se sacará un informe general de los partidos jugados y el resultado final. Además, para cada partido se obtendrá un informe más detallado que incluirá los nombres y nacionalidades de los colegiados que han arbitrado, el nombre de los jugadores que han intervenido en el partido, el puesto en que cada uno ha jugado (que puede ser diferente de su puesto habitual) y los minutos que ha jugado. En este punto hay que señalar que el mismo jugador solo puede intervenir una vez en cada partido, es decir, una vez que el jugador ha salido del campo no puede volver a entrar. En partidos diferentes el mismo jugador puede jugar en distintos puestos.

Otra cuestión a tener en cuenta con respecto a los partidos son los controles antidopaje que se realizan a los jugadores que intervienen en ellos. De cada partido se escogen una serie de jugadores al azar de cada uno de los equipos participantes para realizarle los análisis. Para cada muestra extraída, existe un miembro de la Federación Internacional de Fútbol que se encarga de custodiarla, realizar la tramitación de la muestra y los resultados y un laboratorio encargado de realizar los análisis. La elección de ambos se realiza al azar teniendo en cuenta que un jugador nunca puede tener dos análisis ni con el mismo laboratorio, ni con el mismo miembro de la Federación. Sin embargo, varios jugadores (por ejemplo, todos los de un partido) podrían tener asignado un mismo laboratorio y miembro de la Federación. De cada miembro de la Federación se almacena su nombre, su teléfono y su número de tarjeta identificativa y de los laboratorios, su nombre, dirección, teléfono y CIF.

Al final del campeonato se entregan, además de los premios a las tres primeras selecciones clasificadas, un premio al jugador que más goles ha marcado y otro al jugador que ha marcado el gol más temprano (más cerca del comienzo de un encuentro). Por esta razón es preciso mantener información de qué jugador ha marcado un gol y en qué minuto del partido lo ha marcado.

## Consigna

Según la información dada por el cliente, armá el gráfico de entidad relación.

---

## Caso: La biblioteca II

Imagina que has conseguido un contrato en período de prueba con una empresa de ingeniería de *software* para diseñar los modelos de datos de las aplicaciones.

Esta empresa ha recibido recientemente el encargo de un proyecto por parte de una cadena de bibliotecas privadas establecidas en la ciudad de Madrid. Para probar tu valía profesional, esta empresa te encarga el diseño de una base de datos que recoja lo más fielmente posible los datos manejados por este conjunto de bibliotecas y su semántica. Después de visitar una de estas bibliotecas y conocer su funcionamiento has obtenido las siguientes conclusiones:

- Las bibliotecas que pertenecen a esta organización privada están localizadas en distintos puntos de la ciudad y cada una de ellas suele estar especializada en un área determinada.
- Cada biblioteca dispone de un cierto número de ejemplares del mismo libro y un único ejemplar periódico de cada revista a la que está suscrita.
- A cada biblioteca le está permitido cada cierto tiempo realizar pedidos de ejemplares de libros a una serie de editoriales con las que la organización mantiene acuerdos. La primera vez que un ejemplar de un libro se recibe se cataloga.

Por otra parte, las bibliotecas también pueden suscribirse a revistas publicadas por estas editoriales, de tal manera que a partir de la fecha de suscripción recibirán los ejemplares de manera periódica.

La información que se utiliza para la catalogación de las publicaciones (libros o revistas) es la siguiente:

- Nombre o título de la publicación.
- Temas: conjunto de descriptores de los temas codificados por la organización.
- Editorial: nombre, dirección y teléfono.
- Código ISBN, único para cada publicación.
- Idioma de la publicación.

Además, sobre los libros también se guardan el número de la edición, la fecha de publicación y los autores y de éstos se mantiene, además de su nombre, su nacionalidad. De las revistas se guarda su periodicidad. Hay que señalar que los datos sobre las diferentes publicaciones se encuentran centralizados para todas las bibliotecas y para cada una de ellas lo que se mantiene es la información de los ejemplares de que dispone de cada publicación, sin tener repetidos para cada ejemplar los datos de catalogación. Para identificar los diferentes ejemplares del mismo libro dentro de una biblioteca se utiliza un número de ejemplar y, también, nos interesa la fecha en que fue adquirido.

Para los ejemplares periódicos de las revistas utilizaremos el número real de la revista; por ejemplo, en una determinada biblioteca de la revista AAA a la que la biblioteca está suscrita pueden estar depositados los números 21,22, 23 y 24. Por otra parte, para que una persona pueda retirar un libro de cualquiera de las bibliotecas sólo necesita hacerse socio de una de ellas. Para ello debe hacer una solicitud con sus datos personales (dni, nombre, dirección y teléfono) y a cambio se le entrega un carnet con un número único que le da derecho a tomar prestados libros o revistas de cualquiera de las bibliotecas de la organización. A través de internet y con su número de carnet, y como clave su DNI, el socio puede consultar los títulos disponibles y en qué bibliotecas de la organización puede encontrar ejemplares de los mismos y si éstos están prestados o no. En el futuro se permitirá hacer reservas por la red, y el socio podrá solicitar que un ejemplar de un título concreto esté disponible en la biblioteca que desee, pero esta opción de momento no se quiere implementar. Una vez que la persona es socio de una biblioteca está en disposición de solicitar préstamos de ejemplares, aunque con algunas restricciones. En primer lugar, debe dirigirse a la biblioteca que tiene depositado algún ejemplar del libro o revista que desea solicitar.

En el caso de los libros, los socios pueden tener en préstamo varios libros al mismo tiempo, por un período máximo de una semana. Los ejemplares de las revistas, sin embargo, no pueden ser sacados de las bibliotecas, y un socio no puede tener más de uno de ellos a la vez, del que podrá como máximo disponer durante un día, al final del cual deberá devolverlo. Esta restricción se debe a que sólo se dispone de un ejemplar de cada número periódico de la revista a la que la biblioteca está suscrita. Sólo algunos socios especiales, como investigadores acreditados, pueden saltarse esta restricción y tener en préstamo un ejemplar de una revista con las mismas condiciones que para los libros. Para estos socios, además, debe guardarse el nombre de la entidad para la que trabajan.

Diariamente, un cuarto de hora antes del cierre, en cada biblioteca un proceso nos informa de las revistas en préstamo que deben ser devueltas. Semanalmente, la organización lanza un listado con los ejemplares de libros (y de revistas) que habiendo finalizado su plazo de préstamo aún no han sido devueltos por los socios, con el fin de reclamarlos a éstos.

Anualmente, se obtiene un listado con todos los préstamos de ejemplares de libros, junto con los socios y los períodos en que los han retirado, con el objeto de enviarlos a las editoriales para que éstas construyan sus estadísticas acerca del interés que suscitan sus libros en los lectores. También anualmente se obtienen estadísticas del número de socios nuevos que se han inscrito en cada una de las bibliotecas.

## **Consigna**

Según la información dada por el cliente, armá el gráfico de entidad relación.

A continuación, podrás averiguar las respuestas correctas descargando el siguiente PDF:

## Resolucion



Fuente: elaboración propia.

## Video de habilidades

Es importante saber normalizar nuestros modelos de datos. En este módulo lo que buscamos es poder saber cómo separar y organizar los datos en tablas con atributos y que este modelo que se plantea cumpla al menos con la tercer forma normal, que es hasta donde llegan la mayoría de los diagramas entidad relación.

En el video vemos con desde el ejemplo de pedido proveedor y producto llegamos a armar el DER.

### Video: Normalización de base de datos

Fuente : **Malvarezrc** [malvarezrc] (04/10/2009) *Ejemplo de normalización*. [Youtube]. Recuperado de <https://www.youtube.com/watch?v=-HajWU4pDLM>

## Preguntas de habilidades

La normalización permite obtener estructuras de datos ineficientes

Verdadero

Falso

Justificación

Cuál de las formas normales dice que un registro no puede contener grupo de datos repetitivos.

FN1

FN2

FN3

FN4

Forma normal de Boyce-Codd

Justificación

Si una relación está en FN1 y no tiene claves compuestas quiere decir que también está en ....

FN1

FN2

FN3

FN4
Forma normal de Boyce-Codd
Justificación

En la FN3 los atributos deben depender solo de la clave y de ningún otro atributo de la relación.

Verdadero
Falso
Justificación

Cuando una entidad tiene clave compuesta indica que existe una relación de muchos a muchos

Verdadero
Falso
Justificación

## Cierre

Aprendimos sobre el modelo de datos y sus componentes. Es de suma importancia que como futuro profesional en el área tengas claro todo lo estudiado ya que será de suma importancia para que puedas manipular los datos de forma correcta y ordenada.

## Glosario

## Referencias

**Costal Costa, D.** (2017). Introducción al diseño de bases de datos. Recuperado de <https://pdfslide.net/documents/dolors-costal-costa-introduccion-al-diseno-de-bases-de-datospdf.html>

**Millán, M. E.** (2012). *Fundamentos de bases de datos. Notas de referencia*. Cali, Colombia: G&G Editores.

**Universidad Autónoma del Estado de Hidalgo,** (2017). Diseño de base de datos. Recuperado de <http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/index.html>.