



Conceptos generales de bases de datos

Breve descripción:

En la sociedad de la información es imprescindible el empleo de tecnología, técnicas y procedimientos que, a lo largo de las últimas décadas, han consolidado un marco conceptual importante y necesario en todo proceso de administración de datos, por eso, este componente acerca estos conceptos de manera estructurada para avanzar en los procesos de gestión de la información.

Abril 2024

Tabla de contenido

Introducción	4
1. Conceptos generales de bases de datos.....	5
1.1. Tipos de datos y restricción de no nulidad	10
1.2. Tipos de bases de datos.....	12
1.3. Clasificación de bases de datos	17
1.4. Sistema de gestión de bases de datos.....	18
2. Modelo entidad relación	19
2.1. Relaciones entre entidades	22
Claves	23
2.2. Relaciones de uno a muchos.....	28
REGLA DE MAPEO 1:N de Modelo entidad-relación a modelo relacional	30
2.3. Relaciones de un uno a muchos.....	32
REGLA DE MAPEO N:N de Modelo entidad-relación a modelo relacional....	33
2.4. Relaciones de uno a uno	35
REGLA DE MAPEO 1:1 de Modelo entidad-relación a modelo relacional.....	36
3. Normalización.....	38
3.1. Dependencias funcionales	39
Definición formal DEPENDENCIA FUNCIONAL.....	42

3.2.	Diseño relacional	44
3.3.	Reglas de integridad	51
3.4.	Lenguajes de los sistemas administradores de bases de datos	54
	Sentencias de definición de datos (DDL):	55
	Sentencias de manejo de datos (DML):	56
	Sentencias de control (DCL):	57
	Sentencias de control de transacciones (TCL):	58
4.	Sistema gestor de base de datos	59
	Síntesis	65
	Material complementario	66
	Glosario	67
	Referencias bibliográficas	68
	Créditos	69

Introducción

Una base de datos puede ser concebida como un “almacén” de información, el cual se define y crea una única vez con el objetivo de almacenar grandes cantidades de datos de manera organizada (o estructurada), facilitando así su posterior búsqueda y utilización. En este componente se explicará el concepto y las características de las bases de datos contemporáneas, considerando que cada una es diseñada para satisfacer los requisitos de información específicos de una organización o empresa. Estos diseños están ideados para ser implementados mediante Sistemas de Gestión de Bases de Datos (SGDB), que son sistemas o servicios informáticos que permiten a los usuarios definir, crear, brindar soporte y mantenimiento a las bases de datos, asegurando el control del acceso de manera segura.

1. Conceptos generales de bases de datos

El concepto general de bases de datos según la RAE es:

El conjunto de datos organizado de tal modo que permita obtener con rapidez diversos tipos de información. RAE, 2001.

De acuerdo con esta definición, una hoja de cálculo de Excel puede considerarse una base de datos, o un conjunto de archivos debidamente organizados, o la lista de nombres y teléfonos que está en nuestros “smartphones”. En principio, es correcto llamarle base de datos a estos ejemplos, sin embargo, en el contexto de desarrollo de “software”, se referirá a ese conjunto de información que puede ser almacenada en grandes cantidades de forma organizada y es gestionada desde o a través de un Sistema de Gestión de Bases de Datos (SGBD).

Para organizar y definir la información de forma sistemática, las bases de datos también deben poder almacenar una descripción precisa de los datos que contiene conocida como metadatos, a los que se le relaciona el tipo de información que conceptualmente es guardada (es decir si se agrega una explicación de la naturaleza del dato en la empresa) se da origen a lo que conoce como diccionario de datos o catálogo de datos.

Ejemplo 1

Una empresa gestiona una base de datos de llamadas telefónicas entrantes y salientes de su planta telefónica, en la que se registran los números de teléfono del llamante y la extensión que recibe o realiza la llamada. También se documenta la fecha y hora de cada llamada, si esta fue atendida y su duración, tal como se presenta en la siguiente tabla.

Tabla 1. Base de datos de llamadas telefónicas.

calldate	src	Dst	duration	disposition
2021-04-10 00:12:17.0	3023946875	666	0:05:33	ANSWERED
2021-04-10 00:27:19.0	3023946875	666	0:00:43	ANSWERED
2021-04-10 00:22:52.0	3174377015	666	0:05:40	ANSWERED
2021-04-10 00:24:20.0	3208098045	666	0:08:09	ANSWERED
2021-04-10 00:29:28.0	3162400761	666	0:05:19	ANSWERED
2021-04-10 00:45:23.0	7005	883216361744	0:00:55	ANSWERED
2021-04-10 00:46:59.0	7005	883164829206	0:00:03	ANSWERED
2021-04-10 00:47:38.0	7005	883164829206	0:00:39	ANSWERED
2021-04-10 00:57:04.0	3143259110	666	0:05:23	ANSWERED
2021-04-10 01:29:16.0	7004	0	0:00:08	ANSWERED
2021-04-10 01:31:19.0	22198329	2	0:12:04	ANSWERED
2021-04-10 01:55:44.0	88362500	444	0:06:36	ANSWERED
2021-04-10 02:02:38.0	88362500	666	0:03:01	ANSWERED
2021-04-10 02:14:10.0	3173678610	666	0:01:49	ANSWERED
2021-04-10 02:55:15.0	3184096662	666	0:04:33	ANSWERED
2021-04-10 01:29:30.0	7004	1	1:35:05	ANSWERED
2021-04-10 03:04:30.0	7004	0	0:00:05	ANSWERED
2021-04-10 03:05:35.0	7005	883146610566	0:02:35	ANSWERED
2021-04-10 03:05:41.0	3003264734	666	0:02:56	ANSWERED
2021-04-10 03:18:00.0	7005	147584584	0:00:00	ANSWERED

calldate	src	Dst	duration	disposition
2021-04-10 03:18:50.0	7005	883206657698	0:00:02	ANSWERED
2021-04-10 03:19:06.0	7005	883206657698	0:00:02	NO ANSWERED
2021-04-10 03:19:49.0	7005	883206657698	0:00:00	NO ANSWERED
2021-04-10 03:21:53.0	7005	883113084181	0:00:00	NO ANSWERED
2021-04-10 03:22:40.0	7005	883113084181	0:00:00	NO ANSWERED

La tabla 1 muestra los datos organizados de manera que cada columna agrupa un tipo específico de dato, mientras que cada fila, denominada registro, contiene un conjunto de datos relacionados entre sí. Esta relación indica que cada fila corresponde a una llamada telefónica única. A esta organización se le denomina estructura bidimensional, donde las filas y columnas constituyen las dos dimensiones.

Los metadatos de la tabla anterior serían la definición de cada tipo de dato como se lista a continuación:

calldate: tiene la estructura YYYY-MM-DD HH:MI:SS.Z , donde:

- YYYY sería el año un dato numérico entero de cuatro dígitos.
- MM el mes un dato numérico entero de dos dígitos entre 01 y 12.
- DD un dato numérico entero de dos dígitos entre 01 y 31.
- HH un dato numérico de dos dígitos entre 00 y 23.
- MI y SS un dato numérico de dos dígitos entre 00 y 59.
- Z Un dato numérico de un dígito (para representar milésimas de segundo).

Como este es el conjunto de datos de la primera columna, puede que use un término que describa lo anterior con una sola palabra que es **TIMESTAMP**, de esta forma esa primera columna puede quedar correctamente descrita como:

calldate TIMESTAM (1) Donde 1 es el número de dígitos de milisegundos (Z).

src y dst: empleado para fuente de la llamada o llamante (src) y destino de la llamada (dst) puede ser un número telefónico o de extensión telefónica, algunos tienen el código de país seguido del signo más, ejemplo: 57+3155008002. Por lo tanto, se puede describir como una cadena de texto que no superará los 25 caracteres, pero puede tener menos de 25 caracteres, así el metadato iría descrito:

src VARCHAR(25)

dst VARCHAR(25)

De esta forma decimos que es una cadena de caracteres que varía en longitud de 0 a 25 caracteres.

duration: tiene la estructura HH:MI:SS y como se vio antes representa el tiempo en horas, minutos y segundos:

src TIME TIME se usa para esta estructura de dato en particular

disposition: indica si la llamada es atendida ANSWER, no atendida NO ANSWER, o si falló la FAIL. Se puede representar con una cadena de 10 caracteres.

Por lo tanto, la solución de los metadatos de la base de datos es:

calldate TIMESTAM(1),

src VARCHAR(25),

dst VARCHAR(25),

src TIME,

disposition VARCHAR(10)

Note que cada uno de ellos fue separado por una coma para que se diferencie de los demás.

Ejemplo 2

Conociendo el objetivo de la base de datos de la tabla 1 y los metadatos, se debe crear un diccionario de datos o un catálogo de datos.

Solución

Tabla 2. Diccionario de base de datos de llamadas telefónicas.

Nombre	Creación	Descripción
Base de datos de llamadas telefónicas.	27/02/2021	Registro de las llamadas telefónicas de la PBX de la empresa STD LTDA.

Campo	Tipo de dato	Tamaño	Descripción
calldate	TIMESTAMP	1	Representa el momento exacto en que entra o sale la llamada.
src	VARCHAR	25	Fuente (source) de la llamada, número de quien hace la llamada telefónica.
dst	VARCHAR	25	Número del destino de la llamada telefónica.
duration	TIME	0	Duración de la llamada telefónica.
disposition	VARCHAR	10	Estado final en que se considera quedó la llamada.

Como se presenta en la tabla anterior, un diccionario de datos proporciona detalles sobre la estructura de los datos y cómo se utilizan en la empresa, organización o sistema de información. Para ello, es necesario asignar un nombre al conjunto completo de datos (indicado en la primera fila), establecer una fecha (segunda fila) y proporcionar una descripción (tercera fila) del conjunto de datos. En la tabla 2, se destaca la distinción entre la lógica de los datos (representada por la columna DESCRIPCIÓN del dato) y su almacenamiento físico (indicado por las columnas TIPO DE DATO y TAMAÑO), evidenciando la independencia entre ambos aspectos.

1.1. Tipos de datos y restricción de no nulidad

Existen muchos tipos de datos y varían según el SGDB, ya que cada sistema gestor define sus propios tipos de datos, aunque existen las equivalencias notables y otras no tan notables, se observa una tabla que relacione el tipo de datos según los SGDB más comunes del mercado como ORACLE, PostgreSQL, MySQL y SQLServer.

Tabla 3. Tipos de datos según bases de datos.

Tipo de dato	ORACLE	PostgreSQL	MySQL	SQLServer
Cadena caracteres	VARCHAR2	CHARACTER VARYING	VARCHAR	VARCHAR
Cadena texto	TEXT	TEXT	TEXT	NTEXT
Entero pequeño	SMALLINT	SMALLINT	SMALLINT	SMALLINT
Entero	INTEGER	INTEGER	INT	INT
Fecha	DATE	DATE	DATE	DATE
Fecha y hora	DATE	TIMESTAMP WITH TIME ZONE	DATETIME	.fst- italicDATETIME2

Tipo de dato	ORACLE	PostgreSQL	MySQL	SQLServer
Hora	DATE	TIME	TIME	TIME
Entero con decimales	FLOAT	REAL	FLOAT	FLOAT

Según el tipo de sistema gestor de bases de datos, cada uno define sus tipos de datos que, de alguna forma, son muy parecidos entre ellos, o existen equivalencias, las diferencias son pocas y están relacionadas con aspectos técnicos del almacenamiento, se pueden conocer más detalles de cada uno de ellos en los sitios oficiales de cada motor de base de datos:

Tabla 4. Sistemas de Gestión de Bases de Datos

Nombre	Enlace de acceso
ORACLE	https://docs.oracle.com/cd/E11882_01/server.112/e41085/sqlqr06002.htm#SQLQR959
PostgreSQL	https://www.postgresql.org/docs/13/datatype.html#DATATYPE-TABLE
MySQL	https://dev.mysql.com/doc/refman/8.0/en/data-types.html
SQLServer	https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15

Existen varios tipos de restricciones, por ahora se verán la restricción de longitud, por ejemplo:

src VARCHAR (25)

Se define que src es una columna cuya longitud no puede exceder los 25 caracteres. Asimismo, se establece como obligatorio registrar un dato en dicha columna mediante una restricción de no nulidad, lo que significa que ninguna fila puede tener

un valor nulo en esa columna. Por lo tanto, en el ejemplo de la base de datos de llamadas telefónicas, si se determina que todos los datos son obligatorios mediante el uso de la expresión "NOT NULL", los metadatos que define la estructura de los datos quedaría especificada de la siguiente manera:

```
calldate TIMESTAM(1) NOT NULL,  
  
src VARCHAR(25)NOT NULL,  
  
dst VARCHAR(25)NOT NULL,  
  
src TIME NOT NULL,  
  
disposition VARCHAR(10) NOT NULL
```

Algunos sistemas, para abreviar en la decisión de no nulidad acostumbra a usar el acrónimo NN para representar "NOT NULL".

1.2. Tipos de bases de datos

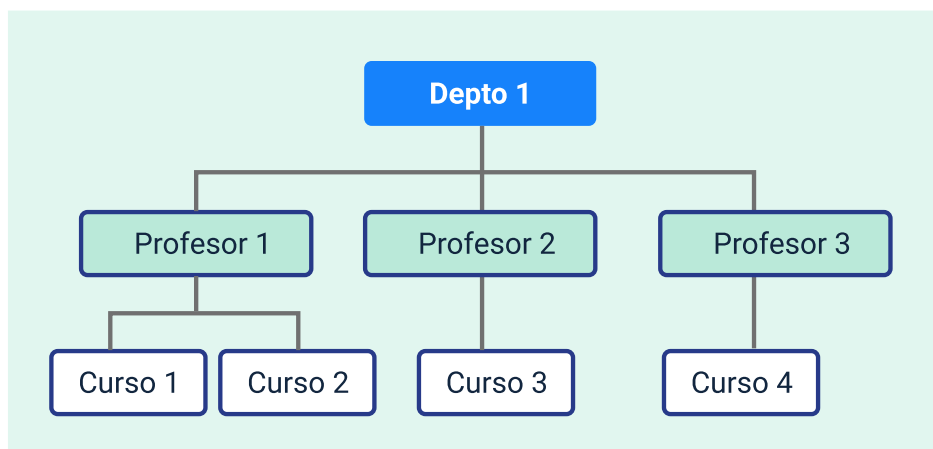
Las bases de datos han evolucionado a lo largo del tiempo, usando diferentes tipos de modelos o enfoques, se ha comprobado que algunos de ellos tienen mejor desempeño en distintos tipos de aplicación. La estructura en la que se configuran los datos al interior de las bases de datos es una forma de clasificación:

- **Bases de datos con estructura jerárquica**

Las primeras bases de datos que se emplearon presentaban una estructura jerárquica, en la que las relaciones entre registros adoptaban la forma de un árbol. Hoy día, un ejemplo representativo de estas bases de datos es LDAP ("Lightweight Directory Access Protocol"), utilizado para gestionar sistemas de directorios que almacenan información variada,

como usuarios, dispositivos, nombres, contraseñas y direcciones, dentro de una red de computadoras.

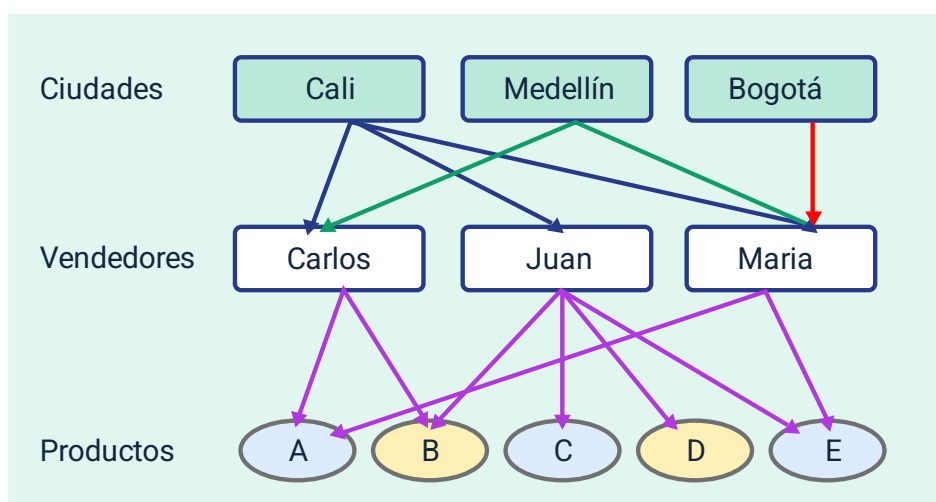
Figura 1. Bases de datos con estructura jerárquica



- **Bases de datos con estructura en red**

Esta estructura de base de datos en red es más compleja que la anterior debido a que no solo admite relaciones jerárquicas, sino que también permite que cada registro establezca conexiones con otros. Así, facilita la existencia de múltiples caminos para acceder a la información.

Figura 2. Bases de datos con estructura en red

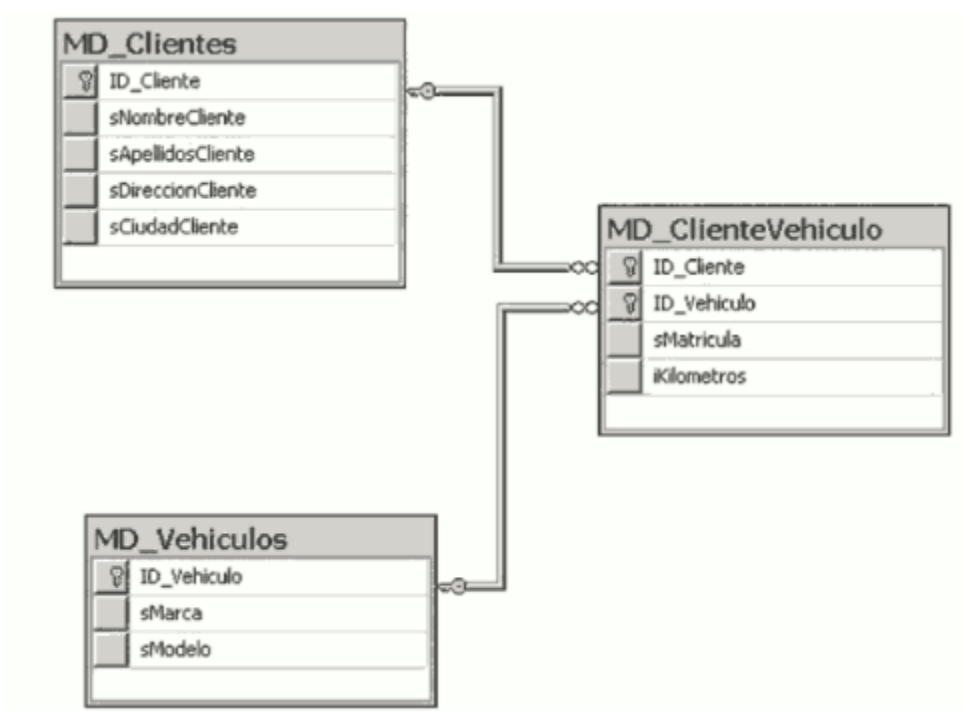


- **Bases de datos con estructura relacional**

Estas bases de datos de estructura relacional son multipropósito, lo que las convierte en las más utilizadas y extendidas en la industria, aunque también son las más complejas de aprender. Por esta razón, nos centraremos únicamente en el diseño e implementación de este tipo de bases de datos.

Se basan en el uso de tablas, similares a las presentadas en la imagen, y en la relación de los datos de unas tablas con otras.

Figura 3. Bases de datos con estructura relacional

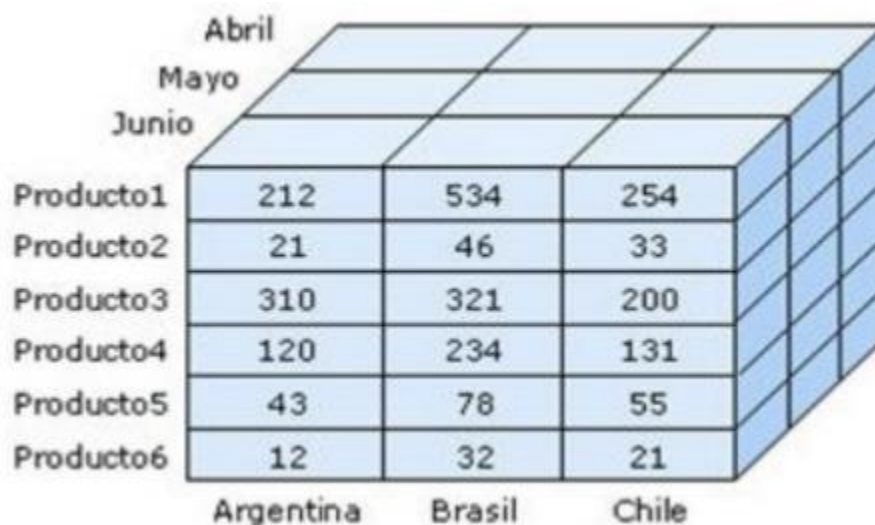


- **Bases de datos con estructura multidimensional**

Es la extensión de las bases de datos de estructuras relacionales, como se vio en las relacionales, se basan en estructuras bidimensionales (tablas),

mientras que la multidimensional en cubos (tres dimensiones), o más complejas con N dimensiones.

Figura 4. Bases de datos con estructura multidimensional

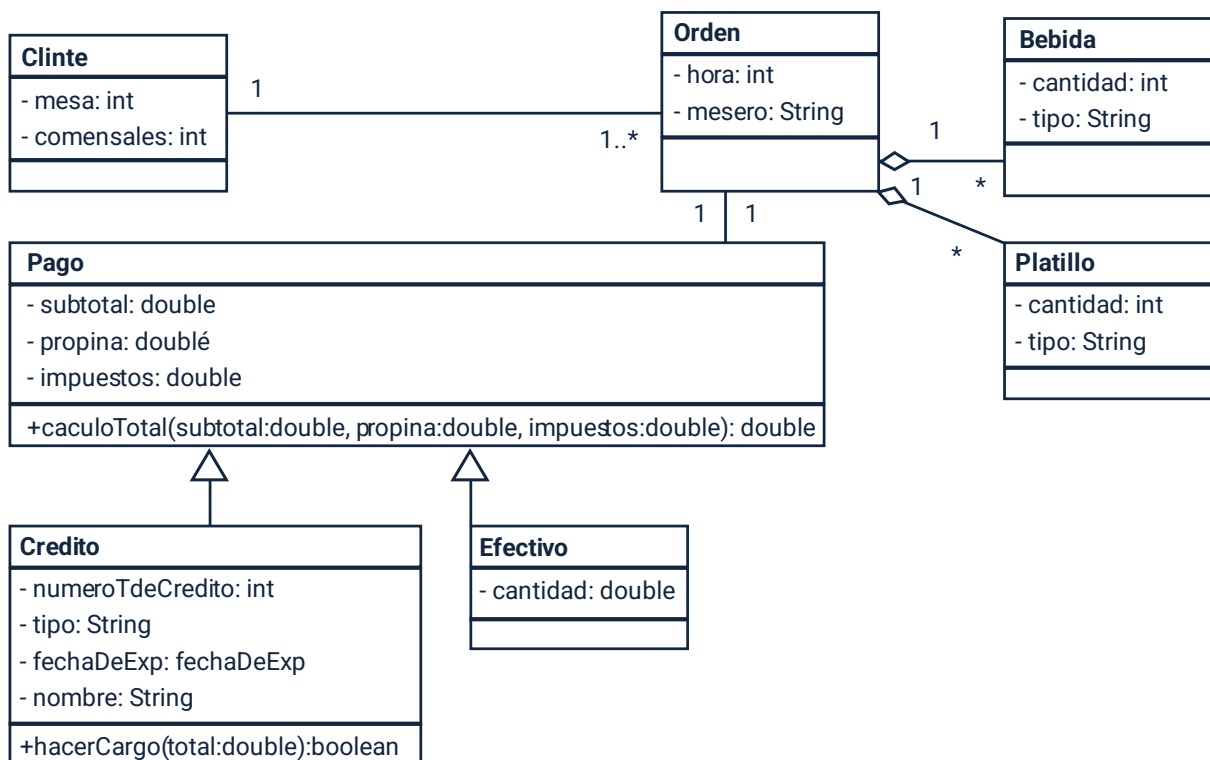


	Abril	Mayo	Junio
Producto1	212	534	254
Producto2	21	46	33
Producto3	310	321	200
Producto4	120	234	131
Producto5	43	78	55
Producto6	12	32	21
	Argentina	Brasil	Chile

- **Bases de datos con estructura orientada a objetos**

Una base de datos orientada a objetos es un sistema de gestión de base de datos mediante el cual representamos la información en forma de objetos que son utilizados en programación orientada a objetos (Kyocera, 2021).

Figura 5. Bases de datos con estructura orientada a objetos



Cada tipo de bases de datos tiene un ámbito de aplicación en el que su desempeño es mejor. Por ejemplo, las bases de datos jerárquicas tienen un mejor desempeño en operaciones de consulta de datos puntuales, mientras que las de datos relacionales son precisas para garantizar la calidad de los datos y la no repetición de los mismos, aunque no muy optimizadas para hacer consultas, y, finalmente, las multidimensionales son muy útiles para análisis estadístico de datos históricos de volúmenes inmensos de datos. Aparte del tipo de datos, existe otro tipo de clasificaciones, ya no según la estructura en que se almacenan los datos, sino de acuerdo con la naturaleza de los datos.

1.3. Clasificación de bases de datos

Según la naturaleza de los datos que se almacenan en la base de datos, se pueden clasificar. La siguiente tabla presenta una compilación de estas clasificaciones.

Tabla 5. Sistemas de Gestión de Bases de Datos

Criterio de clasificación	Clasificación	Descripción
Según la variabilidad de los datos.	Bases de datos estáticas.	<p>Son bases de datos, cuyos datos son históricos, es decir ya no se pueden modificar, se usan comúnmente para estudiar el comportamiento de los datos a través del tiempo.</p> <p>Generalmente, a este tipo de bases de datos se les llama bodega de datos.</p> <p>Muchas de estas bases de datos se almacenan en cubos para su análisis, se las conoce como OLAP, es el acrónimo en inglés de procesamiento analítico en línea (“On-Line Analytical Processing”).</p>
Según la variabilidad de los datos.	Bases de datos dinámicas.	<p>Estos datos se almacenan y pueden ser modificados, agregados, borrados y consultados en cualquier momento, por ejemplo: un sistema de facturación o el sistema Sofiaplus del SENA.</p> <p>A estos sistemas se les denomina transaccionales, porque cada operación de guardar, borrar o editar se configura como una transacción. OLTP es la sigla en inglés de Procesamiento de Transacciones en Línea (“On-Line Transaction Processing”).</p>
Según el contenido.	Bases de datos documentales.	Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes.
Según el contenido.	Base de datos deductivos.	Un sistema de base de datos deductivos es un sistema de base de datos, pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

Pueden existir otras clasificaciones orientadas al ámbito de uso, sin embargo, se presentan las más comunes en el ejercicio del desarrollo de “software”. El dominio de las bases de datos relacionales es esencial en cualquiera de los casos, ya que los conceptos de esta son aplicables y reutilizables en casi todos los otros tipos.

1.4. Sistema de gestión de bases de datos

Un sistema de gestión de la base de datos es un programa de computador que permite definir, crear y mantener los datos de una base de datos, controlando el acceso.

Los servicios SGDB:

- Permiten la definición de la base de datos usando un lenguaje de definición de datos.
- Permiten la inserción, actualización, eliminación y consulta de datos usando un lenguaje de manejo de datos.
- Proporcionan un acceso controlado a la base de datos (con autenticación, roles, niveles de acceso).
- Concurrencia (varios usuarios a la vez accediendo o manipulando los datos) y multitarea.
- Algunos SGDB permiten administrar el catálogo de datos.

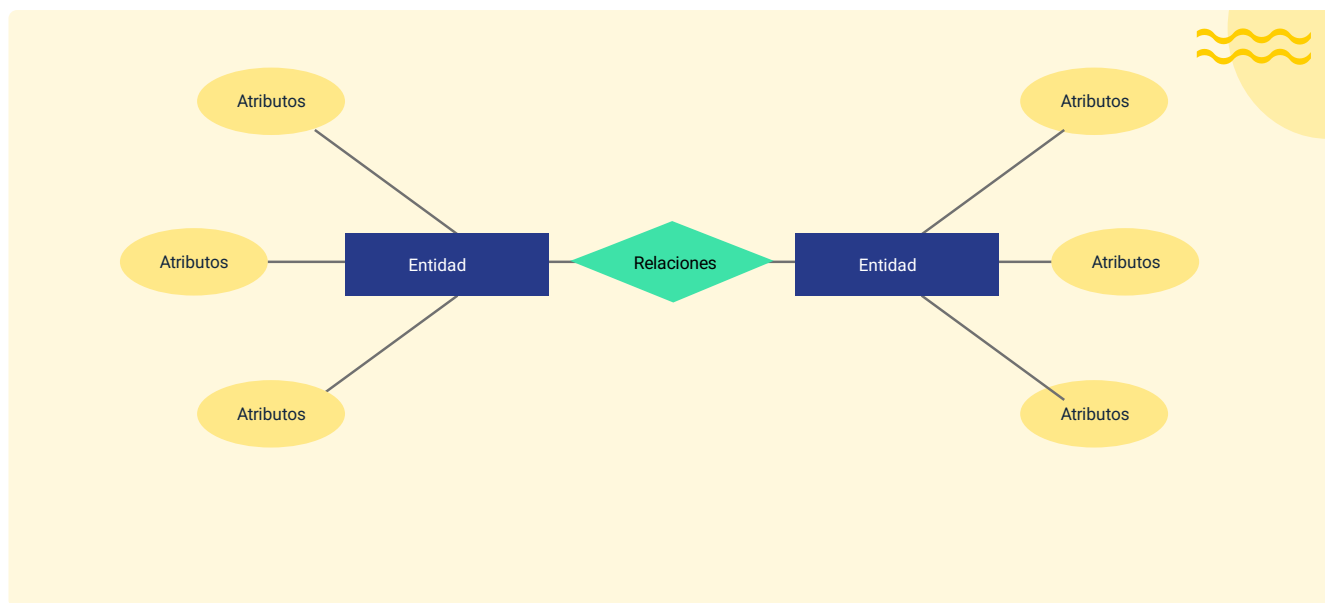
Los SGBD son una herramienta muy útil; sin embargo, se podría decir que los SGBD son un poco más complejos, ya que los usuarios ven más datos y sus relaciones de los que realmente se necesitan.

2. Modelo entidad relación

Las bases de datos relaciones, inicialmente no son fáciles de modelar, por lo que se requiere de cierto nivel de abstracción, es decir, se va ocultando la complejidad del diseño de la base de datos, primero analizando el requerimiento desde un punto de vista conceptual, entendido como las necesidades de la empresa o del sistema de información, luego se va analizando desde el punto de vista de la lógica de los datos y, finalmente, se analiza físicamente como se almacenarán los datos.

Permite representar la información mediante tres elementos fundamentales: entidades (objetos como personas o cuentas, simbolizados con rectángulos), atributos (características de las entidades, representadas con elipses) y relaciones (vínculos entre entidades, ilustrados con rombos). Este modelo proporciona una estructura clara para organizar datos del mundo real, como se presenta a continuación:

Figura 6. Modelo entidad relación



- **Atributos**

Las entidades están descritas por un conjunto de atributos, por ejemplo, una persona puede tener estos atributos: nombres, apellidos, edad, documento de identidad (identificación); o la cuenta bancaria puede tener los siguientes atributos: número de cuenta, saldo, fecha de creación; se representan con una elipse.

- **Entidad**

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de las relaciones entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos. Por ejemplo, cada persona es una entidad, y las cuentas bancarias pueden ser consideradas entidades (Henry, 2002) y se representa con un rectángulo.

- **Relaciones**

Una relación es una asociación entre varias entidades. Por ejemplo, una relación titular asocia un cliente con cada cuenta que tiene (de la que es titular). Se verá más claro en un diagrama, pero antes se deben considerar los siguientes elementos y se representa con un rombo.

Para una base de datos de un sistema bancario, que incluye clientes (es decir, personas) y sus cuentas bancarias, el diagrama Entidad-Relación tiene dos entidades principales: persona y cuenta, cada una con sus atributos específicos.

Se debe tener en cuenta que el modelo relacional convierte las entidades en tablas, por lo tanto, se puede poner como ejemplo que las entidades personas se conviertan en tablas, como se presenta a continuación:

Tabla 6. Persona en una entidad relación

identificación	nombres	apellidos	edad
8.127.346	Ana lis	Mendez	34
212.394.762	Jose Omar	Ciano	28
545.232.546	Luis Dario	Marquez	21
125.446.732	Armando	Bronca Segura	35
33.565.236.787	Vistor José	Diento Paz	24
2.343.554.768	Jose Nicolás	Timado Ruiz	19
45.456.778	Andrea	Martinez	36

Tabla 7. Cuenta en una entidad relación

numero_cuenta	titular	saldo	Fecha_creacion
19.283.746	7	\$ 2.589.954,00	25/08/1998
67.789.901	1	\$ 23.465.212,00	26/08/2001
32.112.312	5	\$ 2.343.242,00	27/08/2004
41.928.374	2	\$ 2.343.423,00	31/08/2013
38.342.657	6	\$ 8.965.211,00	30/08/2010
32.443.171	1	\$ 11.172.594,33	22/06/2001
34.920.608	5	\$ 14.483.578,83	1/09/2016

numero_cuenta	titular	saldo	Fecha_creacion
17.980.455	3	\$ 17.794.563,33	3/09/2019
26.754.825	4	\$ 21.105.547,83	12/04/2022

Las tablas presentan cada una un conjunto de registros, la tabla 6 muestra 6 registros y la tabla 7 muestra 9 de ellas, cada uno de estos registros representa un objeto, es decir, de la tabla persona hay 6 (objetos) o personas, a la agrupación de los datos de cada registro se le llama TUPLA y, en términos sencillos, una TUPLA de una base datos es el conjunto de todos los atributos (columnas) de una fila.

2.1. Relaciones entre entidades

Como se observa en las tablas 6 y 7, una entidad se puede representar con una tabla; sin embargo, la relación titular aún no se sabe cómo representarla y para ello se debe definir algunas propiedades de los atributos.

Principio de unicidad: En una tabla nunca deben existir TUPLAS repetidas, es decir, no debe haber más de una fila con exactamente los mismos valores en cada columna que otra fila de la misma tabla. En otras palabras, cada fila debe diferir al menos en un elemento (columna) en relación con las demás.

En una base de datos relacional, las relaciones permiten evitar datos redundantes. En el ejemplo anterior una persona pudiera tener varias cuentas bancarias, pero no es deseable que, por cada una de las cuentas bancarias asociadas a la misma persona, se deban repetir los datos de esta, sino más bien, que se debe encontrar un mecanismo para relacionar la información de las dos tablas; y para ello se introducen algunos conceptos útiles.

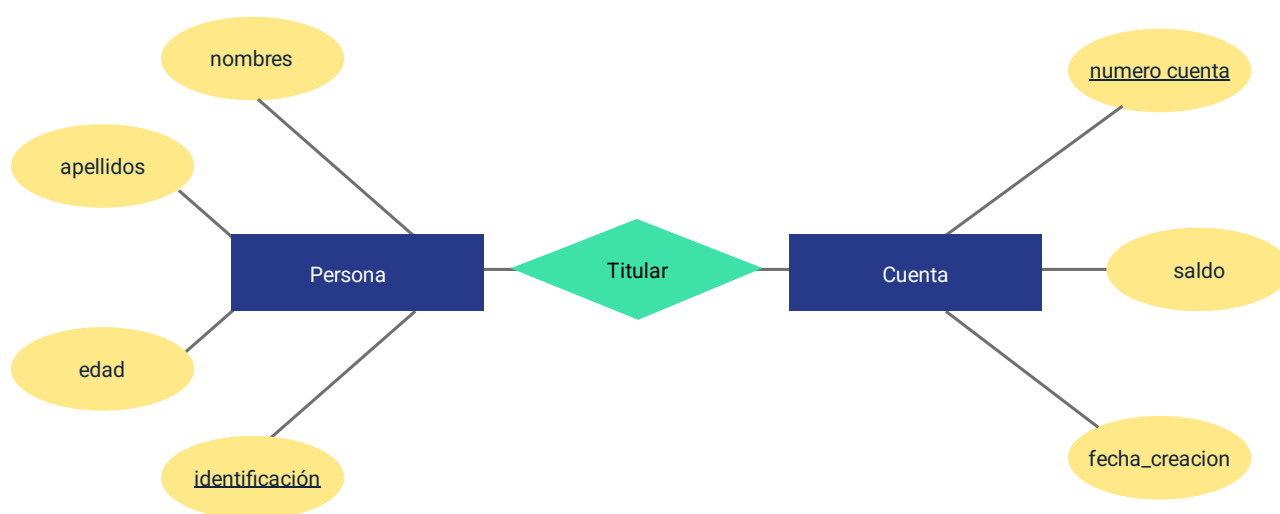
Claves

Dado que en una tabla no hay filas repetidas, se pueden distinguir unas de otras, es decir, se pueden identificar de manera única (por el principio de unidad). Por lo tanto, la manera de identificarlas inequívocamente es según los valores de los atributos. De esta forma, se conoce como claves a los atributos que sirven para identificar un registro de forma inequívoca.

Ejemplo

Para organizar y definir la información de forma sistemática, las bases de datos también deben poder almacenar una descripción precisa de los datos que contiene conocida como metadatos, a los que se le relaciona el tipo de información que conceptualmente es guardada (es decir si se agrega una explicación de la naturaleza del dato en la empresa) se da origen a lo que conoce como diccionario de datos o catálogo de datos.

Figura 7. Claves de las entidades persona y cuenta



Nótese cómo los atributos identificación y numero_cuenta tienen subrayada la palabra identificadora del atributo, representado así que ese atributo es la clave (que hace que se cumpla el criterio de unicidad) de su entidad.

Claves candidata, superclave y clave primaria

Se supone el caso ahora que el banco es internacional, que es un banco moderno y solo diseñado para transacciones vía internet o virtuales, por lo tanto, el correo del cliente (persona) es necesario y obligatorio.

En este caso ocurren un par de asuntos a analizar: al ser internacional el banco, puede existir un cliente en un país con el mismo número de identificación de otro cliente en otro país, por lo tanto, el atributo identificación ya no puede ser una clave de la entidad, sin embargo, es importante y obligatorio saber a qué país corresponde cada cliente, el conjunto de datos identificación y país y si ambos cumplen con el criterio de unidad, porque no existen dos personas en el mismo país con el mismo número de identificación.

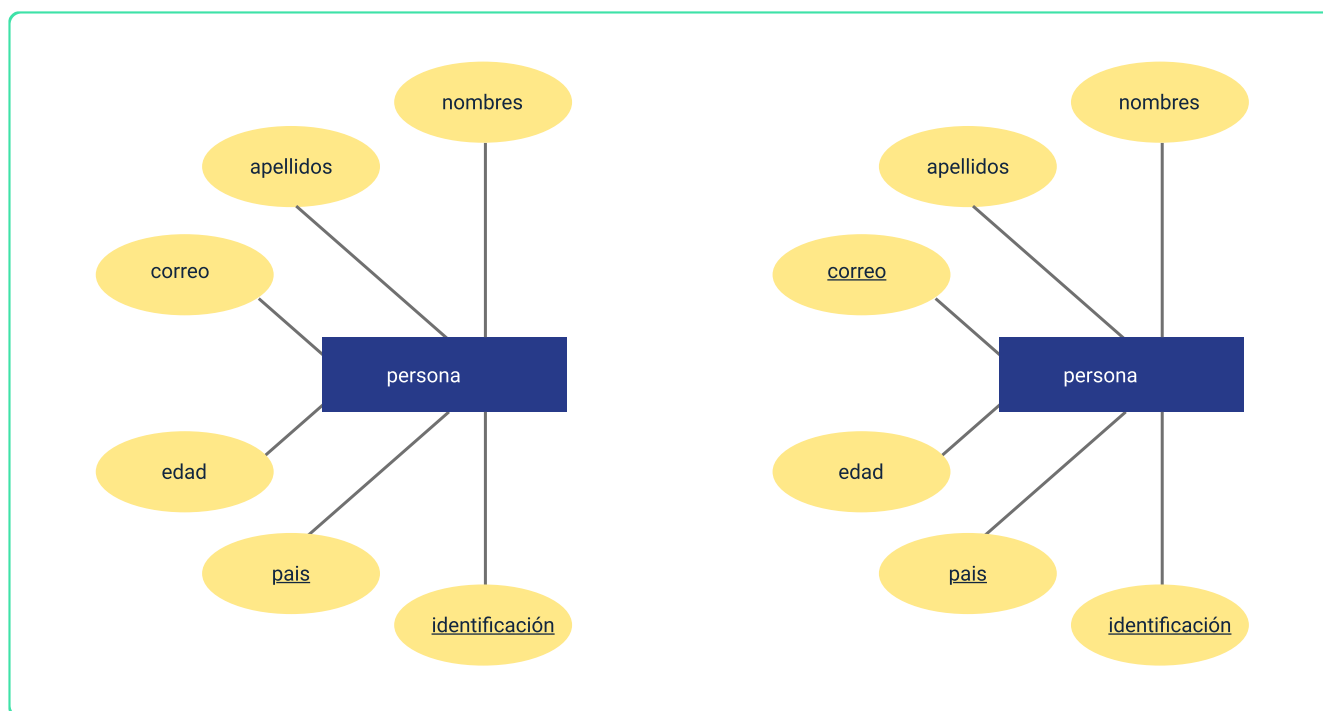
Clave compuesta

Si se escogiera el conjunto de atributos identificación y país como clave de la entidad, se está haciendo una clave compuesta porque es un conjunto de más de un atributo el que se usa para este fin.

El otro aspecto a considerar es que se puede utilizar el correo electrónico como clave, dado que no existen dos usuarios con el mismo correo electrónico. Esto es válido si se estableció como regla del banco que cada cliente debe tener un correo electrónico distinto al de los demás.

Estos dos casos se ilustran en la figura siguiente.

Figura 8. Claves candidatas



Clave candidata

Los dos caminos propuestos en la figura 8 son válidos, y se denominan claves candidatas porque con ambas se puede lograr el principio de unicidad, y es potestad de cada diseñador o analista de sistemas determinar cuál de las dos emplear.

Existe otro tipo de solución y es asignarle un atributo identificador, a la entidad persona que se encargue exclusivamente de identificar inequívocamente cada tupla o fila, a este atributo se le puede llamar id_persona que sería un número que el programa o “software” para el que se está desarrollando la base le asignará a cada cliente, cada vez que se registre por primera vez, los datos de una persona (generalmente una secuencia).

De esta forma, se tendrían 3 claves candidatas; la figura 9 muestra cómo sería la entidad y la tabla que representa.

Figura 9. Claves candidatas secuencia

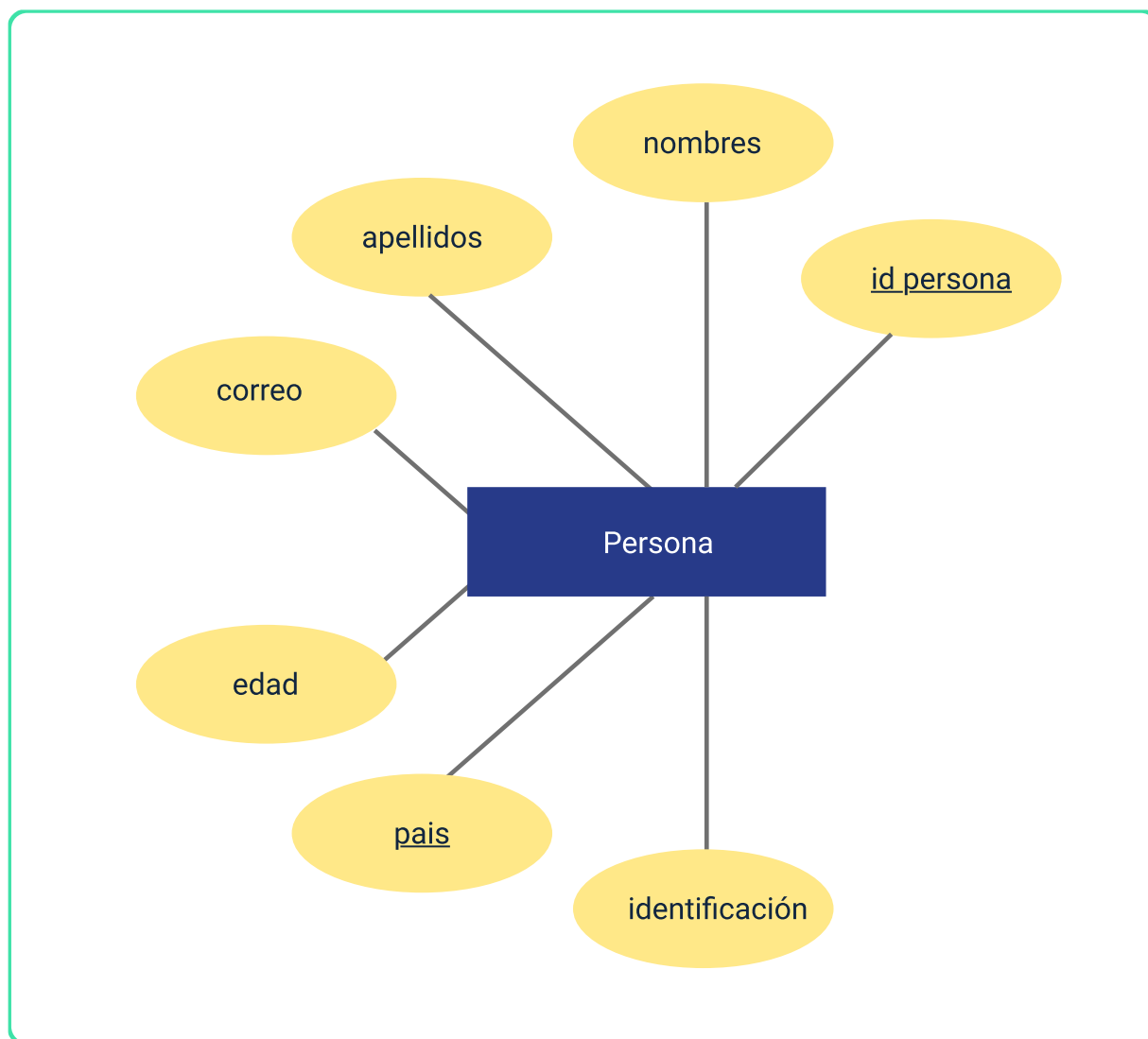


Tabla 8. Entidad persona

id_persona	pais	identificación	correo	nombres	apellidos	edad
1	colombia	8.127.346	ana@rmail.com	Ana lis	Mendez	34

id_persona	pais	identificacion	correo	nombres	apellidos	edad
2	colombia	212.394.762	jose@pmail.com	Jose Omar	Ciano	28
3	colombia	545.232.546	luis@jmail.com	Luis Dario	Marquez	21
4	colombia	125.446.732	arm@hmail.com	Armando	Bronca Segura	35
5	colombia	33.565.236.787	victor@umail.com	Vistor José	Diento Paz	24
6	colombia	2.343.554.768	josen@pmail.com	Jose Nicolás	Timado Ruiz	19
7	peru	45.456.778	andrea@email.com	Andrea	Martinez	36

Este último enfoque es el más empleado y recomendado por los diseñadores o analistas de “software”, pues reduce, significativamente, la complejidad del diseño y facilita la relación con otras tablas o entidades. Por lo tanto, se usará cada vez que se tenga oportunidad de hacerlo.

Superclave

Es un conjunto de uno o más atributos que, tomados juntos, permiten identificar de forma única una entidad en el conjunto de entidades. Por ejemplo, el atributo id_persona del conjunto de entidades persona es suficiente para distinguir una persona cliente del banco de las otras. Así, id_persona es una superclave. Análogamente, la combinación de identificación y país es una superclave del conjunto de entidades persona. El atributo correo de persona es otra superclave, pero el atributo nombres no lo es porque varias personas podrían tener el mismo nombre.

Clave primaria

Se emplea el término clave primaria para nombrar una clave candidata que es elegida por el diseñador o analista de la base de datos como clave principal para

identificar las tuplas dentro de una entidad. Una clave (primaria, candidata y superclave) es una propiedad del conjunto de entidades, más que de las entidades individuales. Entre dos tuplas individuales en el conjunto no pueden tener el mismo valor en sus atributos clave primaria al mismo tiempo. La designación de una clave primaria es en sí una restricción que modela una característica del mundo real.

2.2. Relaciones de uno a muchos

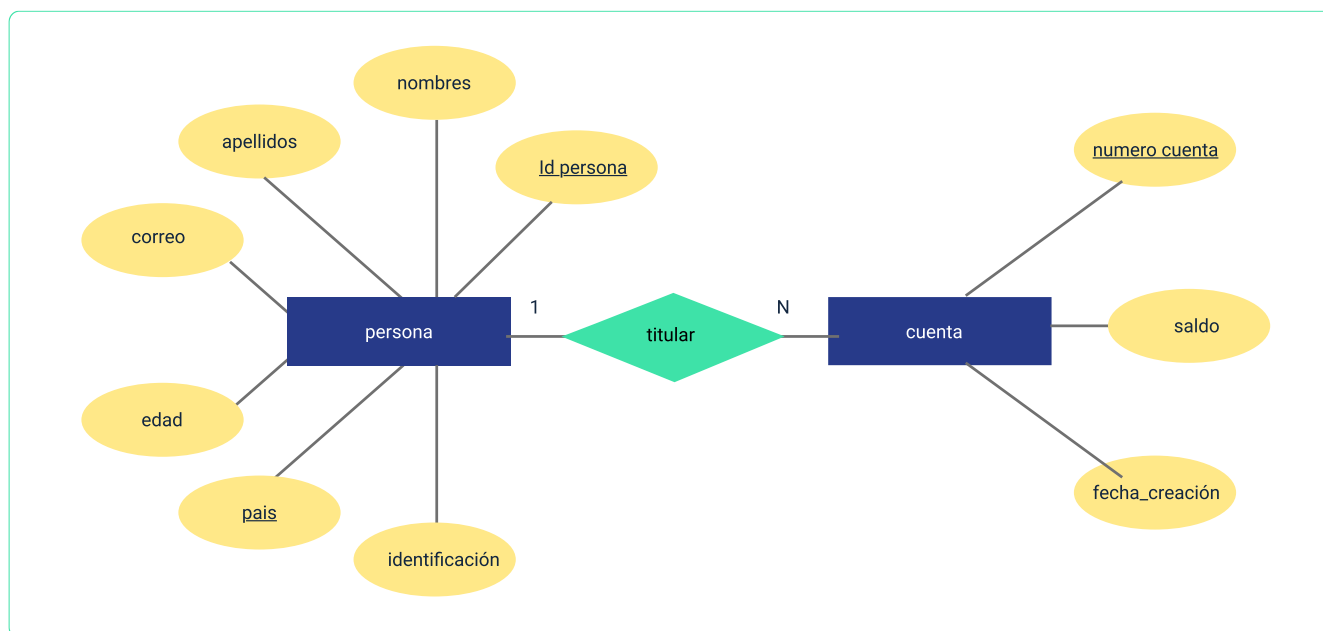
Una relación de uno a varios o de uno a muchos es el tipo de relación más empleada. En este tipo de relación, una fila de la tabla A puede tener muchas filas coincidentes en la tabla B. Pero una fila de la tabla B solo puede tener una fila coincidente en la tabla A.

Ejemplo

Las tablas persona y cuenta tienen una relación de uno a muchos. Es decir, cada persona puede tener más de una cuenta, pero, por ejemplo, cada título proviene de una sola editorial.

En un diagrama de entidad relación esto se representa con lo que se denomina multiplicidad como se muestra en la siguiente figura.

Figura 10. Relaciones de uno a muchos



A esta representación se le denomina multiplicidad 1:N y se lee de la siguiente forma: UNA persona es titular de N cuentas bancarias, de esta forma una persona puede tener N (que puede ser cero, uno o más de una) cuentas bancarias. Se puede leer en forma inversa que el resultado es el mismo: muchas o varias cuentas bancarias pueden tener un mismo titular.

Cuando una relación es de uno a muchos y se mapea una un modelo de base de datos relacional, la relación titular (rombo) se termina convirtiendo una columna de la tabla de los muchos.

Figura 11. Tablas de un modelo relacional 1:N

id_persona	pais	identificacion	correo	nombres	apellidos	edad	numero_cuenta	titular	saldo	fecha_creacion
1	colombia	8.127.346	ana@rmail.com	Ana lis	Mendez	34	19.283.746	7	\$ 2.589.954,00	25/08/1998
2	colombia	212.394.762	jose@pmail.com	Jose Omar	Ciano	28	67.789.901	1	\$ 23.465.212,00	26/08/2001
3	colombia	545.232.546	luis@jmail.com	Luis Dario	Marquez	21	32.112.312	5	\$ 2.343.242,00	27/08/2004
4	colombia	125.446.732	arm@hmail.com	Armando	Bronca Segura	35	41.928.374	2	\$ 2.343.423,00	31/08/2013
5	colombia	33.565.236.787	victor@umail.com	Vistor José	Diento Paz	24	38.342.657	6	\$ 8.965.211,00	30/08/2010
6	colombia	2.343.554.768	josen@pmail.com	Jose Nicolás	Timado Ruiz	19	32.443.171	1	\$ 11.172.594,33	22/06/2001
7	perú	45.456.778	m	Andrea	Martinez	36	34.920.608	5	\$ 14.483.578,83	1/09/2016
							17.980.455	3	\$ 17.794.563,33	3/09/2019
							26.754.825	4	\$ 21.105.547,83	12/04/2022

a) Tabla persona

b) Tabla cuenta

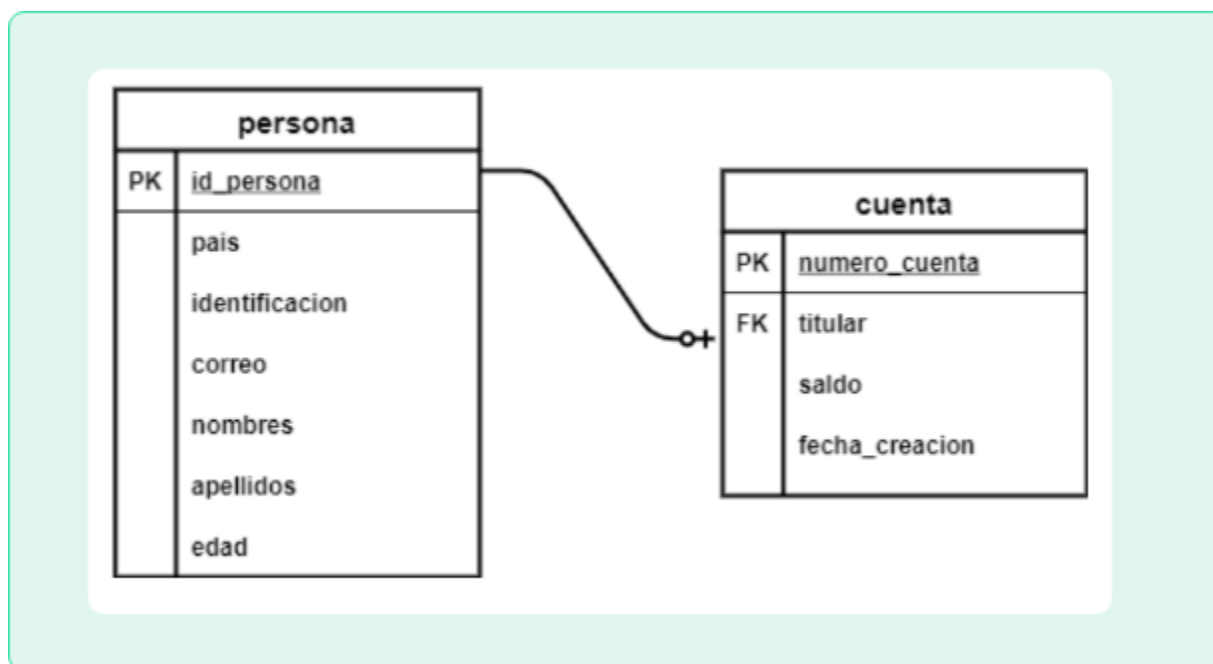
Como se evidencia en la figura 11, Ana Lis Méndez es titular de dos cuentas bancarias (67.789.901 y 32.443.171). Todos los datos en la columna titular de la tabla cuenta se denominan LLAVE FORÁNEA de la tabla persona.

REGLA DE MAPEO 1:N de Modelo entidad-relación a modelo relacional

Una relación de uno a mucho (1:N) se transforma en una columna en la tabla que tiene la multiplicidad de los muchos (cuenta). Y todos los datos de esta columna existen como llave primaria en la tabla que tiene la multiplicidad uno (tabla persona).

Existe otro tipo de diagrama que es más utilizado al diseñar una base de datos; este diagrama se conoce como "Diagrama relacional", en el cual las entidades se presentan directamente como tablas y las relaciones son más evidentes. A continuación, se presenta cómo se transforma la representación del diagrama entidad-relación de la figura en un diagrama relacional.

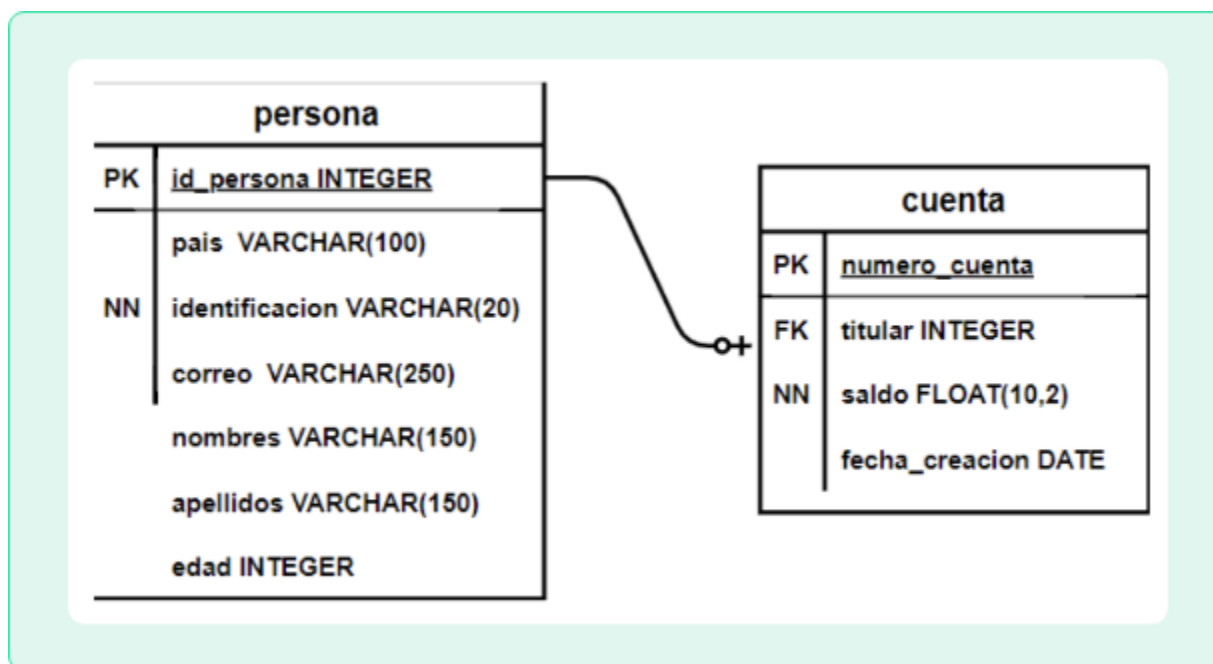
Figura 12. Diagrama relacional 1:N



En la figura 12 se puede identificar el nombre de las entidades (persona y cuenta), los atributos haciendo especial énfasis en los que representan la llave primaria con el indicador PK (del inglés “Primary Key”), y también se observa la relación titular (columna de la tabla), indicando que es una clave o llave foránea de la tabla persona, lo indica con el indicador FK (del inglés “Foreign Key”).

Algunos diagramas relacionales agregan metadatos para hacer más claro el modelo al diseñador, así como la naturaleza de los datos del modelo.

Figura 13. Diagrama relacional 1:N con metadatos



Conclusión: la relación de uno a muchos se termina convirtiendo en una columna en la tabla que tiene la multiplicidad de los muchos.

2.3. Relaciones de un uno a muchos

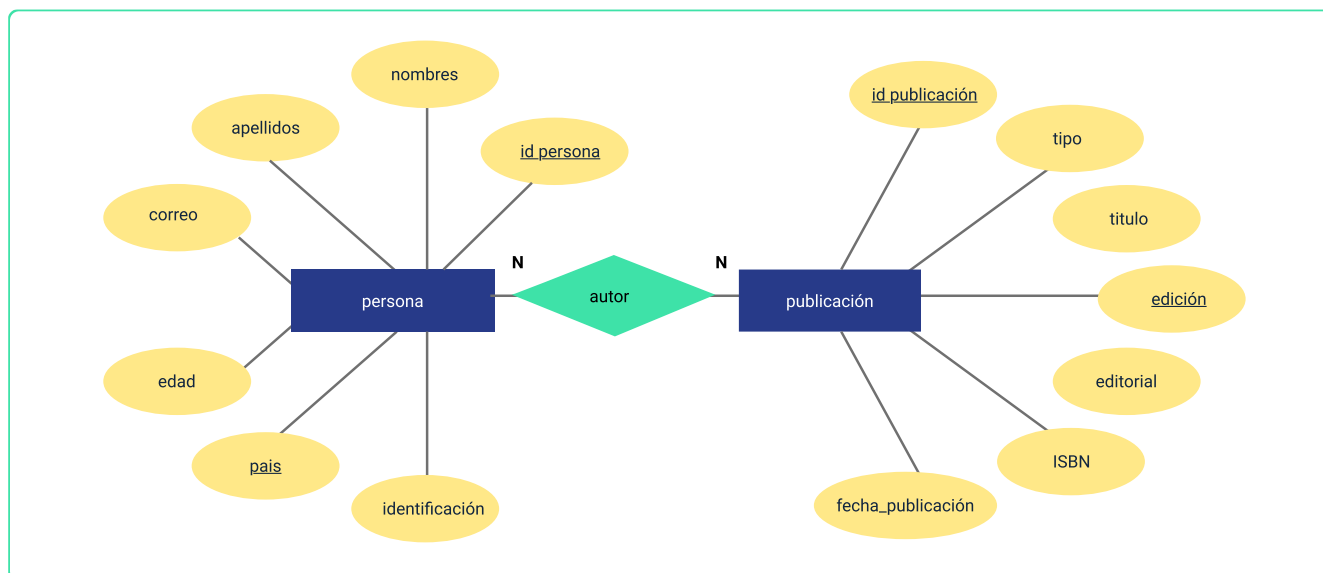
En una relación de muchos a muchos, una fila de la tabla A puede estar relacionada con varias filas de la tabla B y, de manera inversa, una fila de la tabla B puede relacionarse con varias filas de la tabla A. Esta relación se establece mediante la creación de una tercera tabla, conocida como tabla de relación. Esta nueva tabla posee una clave principal compuesta por las claves foráneas tanto de la tabla A como de la B.

Ejemplo

Se necesita almacenar la información de unas referencias bibliográficas o publicaciones de libros, tesis, artículos que escriben los estudiantes de una universidad, y como se sabe, un libro, tesis, artículo o cualquier publicación puede ser elaborada por

más de una persona, y a su vez, una persona (docente, estudiante o investigador) puede tener más de una publicación de su autoría.

Figura 14. Diagrama de entidad de publicaciones bibliográficas



Como se identifica en la figura 14, la entidad publicación tiene una llave primaria llamada identificación, y existe una relacionada llamada autor, y representa que una persona pueden ser autor de una o muchas publicaciones, y que una publicación puede ser de autoría de una o muchas personas, la multiplicidad en este caso es de muchos a muchos y se representa como N:N.

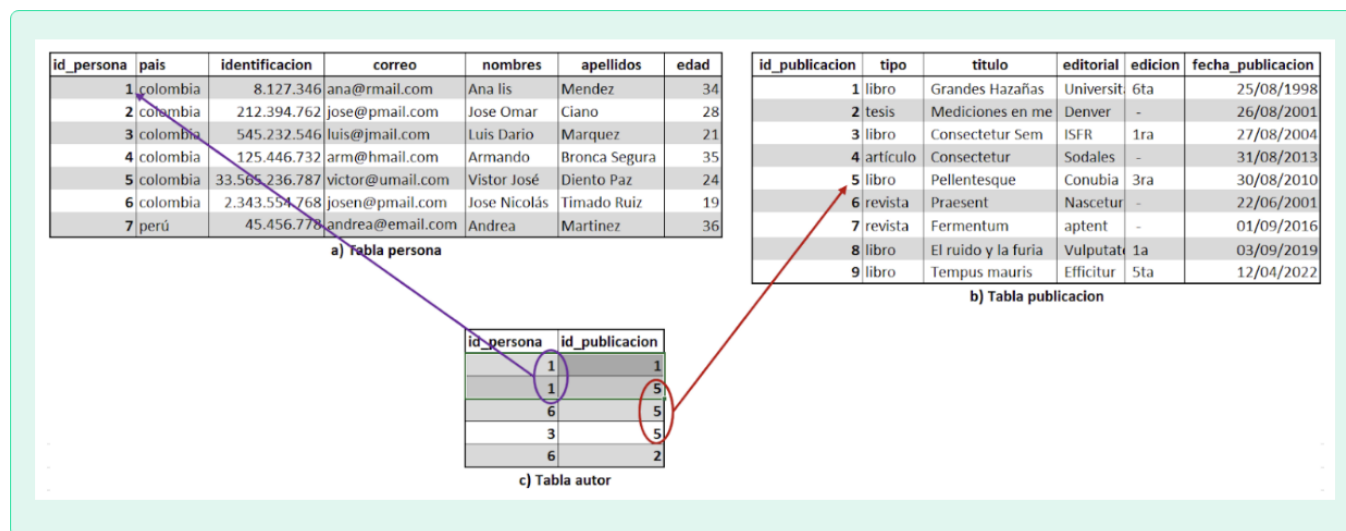
REGLA DE MAPEO N:N de Modelo entidad-relación a modelo relacional

En una relación de uno a mucho (N:N) se transforma en una tabla cuya llave primaria está compuesta por las llaves primarias de las otras dos tablas.

En la figura 15, la “tabla autor”, dice que la persona Ana Lis, es autora de los libros de título Grandes hazañas y Pellentesque, también que el libro Pellentesque es

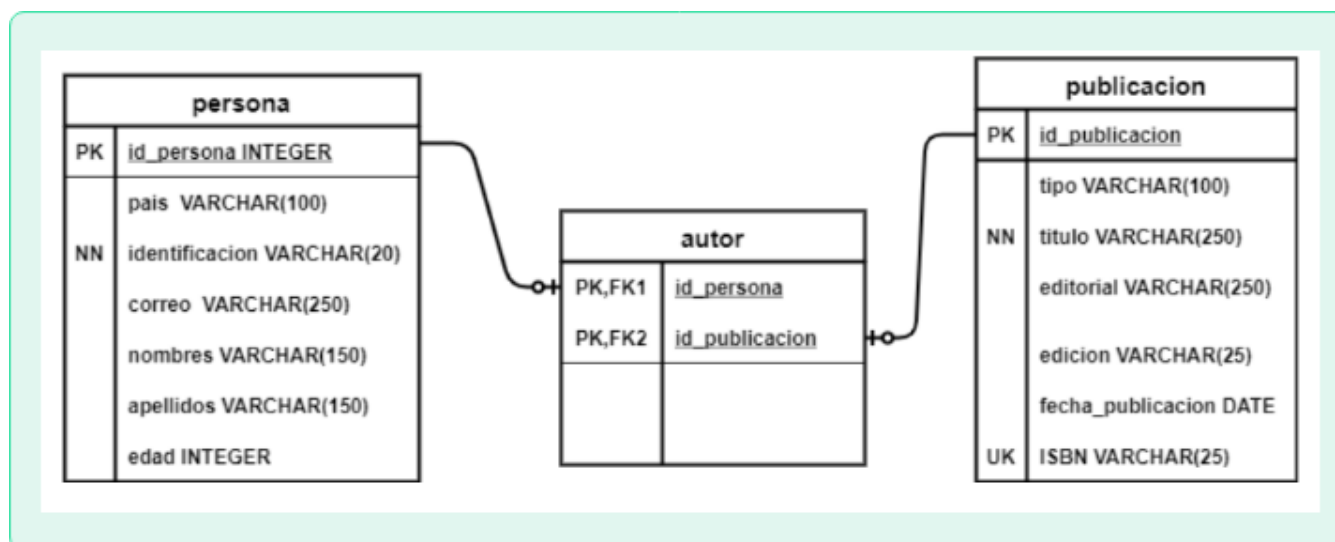
escrito además de Ana Lis por Luis Darío y Jose Nicolás. Lo anterior haciendo referencia a las llaves primarias de las tablas a) persona y b) publicacion.

Figura 15. Tablas de relación N:N.



La representación de la relación N:N con un diagrama relacional se muestra en la figura 9. Donde se usan los prefijos FK1 y FK2, para indicar llave foránea uno y dos, para las entidades persona y publicación 2, respectivamente. También se puede identificar el prefijo UK ("Unique Key") en la tabla publicación, en la columna ISBN, el cual significa que en toda la tabla publicación no puede existir más de una fila con el mismo ISBN es decir, nunca ninguna publicación tendrá el mismo ISBN que otra.

Figura 16. Tablas de relación N:N.



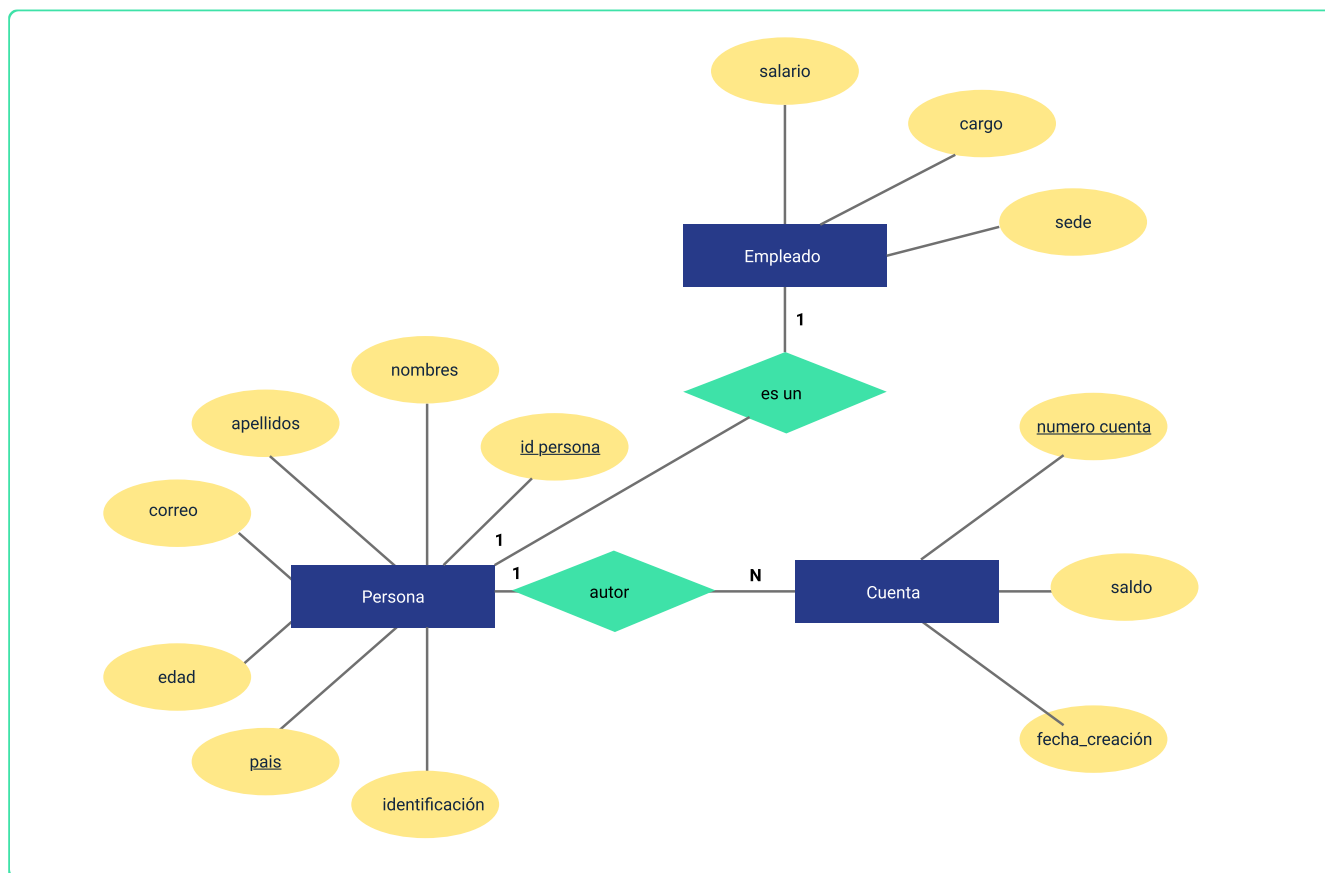
2.4. Relaciones de uno a uno

En una relación uno a uno, una fila de la tabla A solo puede tener una fila coincidente en la tabla B y viceversa. Se crea, entonces, una relación uno a uno si ambas columnas relacionadas son claves principales o tienen restricciones únicas. Este tipo de relación no es común porque la mayoría de la información que está relacionada de esta manera estaría en una tabla; se puede usar una relación uno a uno por alguno de estos motivos:

- Dividir una tabla con muchas columnas.
- Aislar parte de una tabla por motivos de seguridad.
- Almacenar datos de corta duración y que podrían eliminarse fácilmente borrando la tabla periódicamente.
- Almacenar información que solo se aplique a un subconjunto de la tabla principal.

Volviendo al ejemplo del banco, esta vez se supone que la empresa necesita información sobre sus empleados, lo que, a la vez, pueden ser clientes del banco.

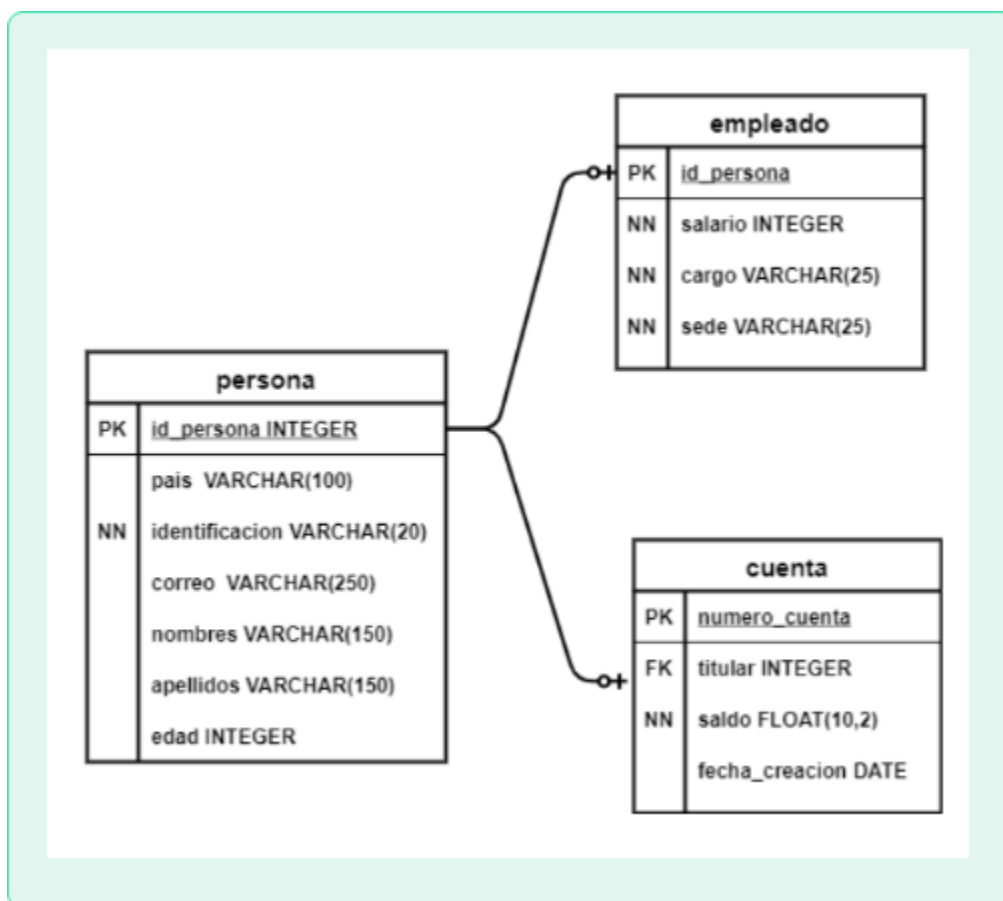
Figura 17. Diagrama entidad relación persona-empleado



REGLA DE MAPEO 1:1 de Modelo entidad-relación a modelo relacional

En el momento de convertir la relación o con multiplicidad 1:1 a tablas, la relación uno a uno se convierte en una columna de la tabla empleado con restricción de unicidad ("Unique Key" o "Primary Key") como se muestra en el siguiente diagrama relacional.

Figura 18. Diagrama relacional 1:1



Como se puede inferir de la figura 18, se crea una tabla donde se relacionan los datos de las personas que son empleados del banco, a través de una columna que se llama `id_persona`, pero es común entre algunos diseñadores ponerle el nombre que lo relacione con la tabla a la que pertenece, es decir, `id_empleado`, pero esto queda al criterio del diseñador.

3. Normalización

La normalización es el procedimiento mediante el cual se aplican las reglas de mapeo o conversión de un modelo entidad-relación a un modelo relacional, es decir a tablas y relaciones, los principios rectores de este proceso son: no redundancia de datos (que no se repitan los datos) y la dependencia de los datos dependencia incoherente (una separación lógica de datos en tablas).

Los datos redundantes desperdician espacio en disco y crean problemas de mantenimiento. Por ejemplo, en la figura 18 se ve que los datos comunes de los clientes y los bancos están en la tabla persona, mientras que los datos que pertenecen exclusivamente a los empleados están en otra tabla que se relaciona con la de personas.

¿Qué es una “dependencia incoherente”?

Aunque resulta intuitivo para un usuario buscar en la tabla de persona el correo de un cliente en particular, es posible que no tenga sentido buscar en esa columna el salario del empleado que se relaciona con esos datos de la tabla persona, pues el salario del empleado está relacionado con la tabla empleado o depende de él. Las dependencias incoherentes pueden dificultar el acceso a los datos, ya que la ruta para encontrar los datos puede faltar o estar dañada.

Existen unas reglas básicas de normalización, las más comunes son las 3 formas normales que se listan a continuación y se desarrollarán con un ejemplo más complejo.

Primera forma normal

- Eliminar grupos de repetición en tablas individuales.
- Crear una tabla independiente para cada conjunto de datos relacionados.

- Identificar cada conjunto de datos relacionados con una clave primaria.

Segunda forma normal

- Crear tablas independientes para conjuntos de valores que se aplican a varios registros.
- Relacionar estas tablas con una clave foránea.

Tercera forma normal

- Eliminar los campos que no dependen de la clave.

Hay una cuarta forma normal, también llamada forma normal de “Boyce Codd” (BCNF), y una quinta forma normal, pero rara vez se consideran en un diseño práctico, solo agregan complejidad al sistema y no aportan un valor funcional que aporte a la solución del problema, por esta razón no será tratada.

3.1. Dependencias funcionales

Una dependencia funcional es un tipo de restricción que termina construyendo una generalización del concepto de clave, como se estudió en el modelo E-R y en el modelo relacional. Pero no es tan fácil localizar las dependencias, ya que necesitan de un análisis de los atributos (columnas) o, con más precisión, de las interrelaciones entre atributos y, frecuentemente, la intuición no es suficiente a la hora de encontrar y clasificar todas las dependencias. Aunque existe una teoría matemática para realizar este análisis, un ejemplo sencillo puede enseñar cómo analizar estas dependencias de manera intuitiva, tal como se presenta a continuación.

Ejemplo

Un cliente pide que desarrolle un “software” para llevar el inventario de una ferretería, los productos, los proveedores, el precio al que cada proveedor vende cada producto, para lo cual suministra la siguiente tabla, y se debe identificar las dependencias funcionales para poder realizar un modelo relacional.

Tabla 9. Proveedores, productos y precios

nit_proveedor	Correo	Nombres	Telefono	Codigo_producto	Precio
8127346	ana@rmail.com	Ana Lis Mendez	398589854	Masillas FFC	\$ 25.890,00
212394762	jose@pmail.com	Jose Omar	5475698	Pistola de silicona	\$ 14.300,00
545232546	luis@jmail.com	Luis Dario	3984558	Cutter RDS	\$ 8.700,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	Lija C20	\$ 820,00
212394762	jose@pmail.com	Jose Omar	5475698	Masillas FFC	\$ 26.000,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	Lija C20	\$ 7.900,00
545232546	luis@jmail.com	Luis Dario	36584114	Pistola de silicona	\$ 12.990,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	Lija C20	\$ 830,00
212394762	jose@pmail.com	Jose Omar	5475698	Cutter RDS	\$ 9.200,00

El principio de no repetición de datos sugiere que se deben identificar todos los datos que se repiten. Esto se puede conocer en la siguiente figura. Es importante

destacar que los valores de nit_proveedor son iguales en todos los registros que comparten el mismo valor de correo, nombres y teléfono.

Figura 19. Dependencia de nit_proveedor con datos del proveedor

nit_proveedor	correo	nombres	telefono	codigo_producto	producto	precio
8127346	ana@rmail.com	Ana Lis Mendez	398589854	MS-028	Masillas FFC	\$ 25.890,00
212394762	jose@pmail.com	Jose Omar	5475698	PS-012	Pistola de silicona	\$ 14.300,00
545232546	luis@jmail.com	Luis Dario	3984558	CT-003	Cutter RDS	\$ 8.700,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 820,00
212394762	jose@pmail.com	Jose Omar	5475698	MS-028	Masillas FFC	\$ 26.000,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 7.900,00
545232546	luis@jmail.com	Luis Dario	36584114	PS-012	Pistola de silicona	\$ 12.990,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 830,00
212394762	jose@pmail.com	Jose Omar	5475698	CT-003	Cutter RDS	\$ 9.200,00

También en la figura 20 la columna código_producto es la misma para la columna producto, pero difiere del precio que cada proveedor ofrece del mismo.

Figura 20. Dependencia de código_producto_ con datos del producto

nit_proveedor	correo	nombres	telefono	codigo_producto	producto	precio
8127346	ana@rmail.com	Ana Lis Mendez	398589854	MS-028	Masillas FFC	\$ 25.890,00
212394762	jose@pmail.com	Jose Omar	5475698	PS-012	Pistola de silicona	\$ 14.300,00
545232546	luis@jmail.com	Luis Dario	3984558	CT-003	Cutter RDS	\$ 8.700,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 820,00
212394762	jose@pmail.com	Jose Omar	5475698	MS-028	Masillas FFC	\$ 26.000,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 7.900,00
545232546	luis@jmail.com	Luis Dario	36584114	PS-012	Pistola de silicona	\$ 12.990,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 830,00
212394762	jose@pmail.com	Jose Omar	5475698	CT-003	Cutter RDS	\$ 9.200,00

La dependencia código_producto no se cumple para el precio del producto, por lo tanto, dice que a cada proveedor y producto debe existir un precio distinto como se identifica en la siguiente figura.

Figura 21. Dependencia de nit_proveedor, código_producto_ con precio del producto

nit_proveedor	correo	nombres	telefono	codigo_producto	producto	precio
8127346	ana@rmail.com	Ana Lis Mendez	398589854	MS-028	Masillas FFC	\$ 25.890,00
212394762	jose@pmail.com	Jose Omar	5475698	PS-012	Pistola de silicona	\$ 14.300,00
545232546	luis@jmail.com	Luis Dario	3984558	CT-003	Cutter RDS	\$ 8.700,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 820,00
212394762	jose@pmail.com	Jose Omar	5475698	MS-028	Masillas FFC	\$ 26.000,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 7.900,00
545232546	luis@jmail.com	Luis Dario	36584114	PS-012	Pistola de silicona	\$ 12.990,00
8127346	ana@rmail.com	Ana Lis Mendez	398589854	LG-021	Lija C20	\$ 830,00
212394762	jose@pmail.com	Jose Omar	5475698	CT-003	Cutter RDS	\$ 9.200,00

En la siguiente tabla se presentan las dependencias detectadas.

Tabla 10. Dependencias funcionales

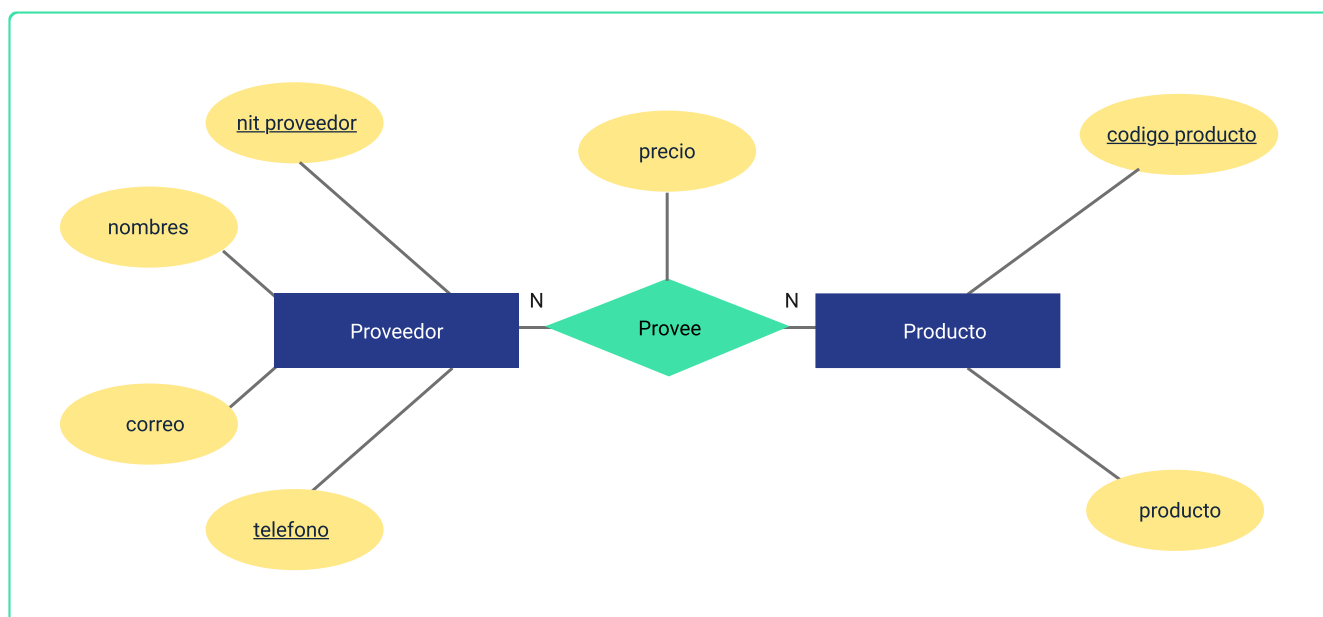
Atributo	Es dependencia funcional de:
nit_proveedor	correo nombres teléfono
codigo_producto	producto
nit_proveedor, codigo_producto	precio

Definición formal DEPENDENCIA FUNCIONAL

Se dice que un atributo X de una relación “depende funcionalmente” de otro atributo o conjunto de atributos Y de la relación si a todo valor (o valores del conjunto) Y le corresponde siempre el mismo valor de X.

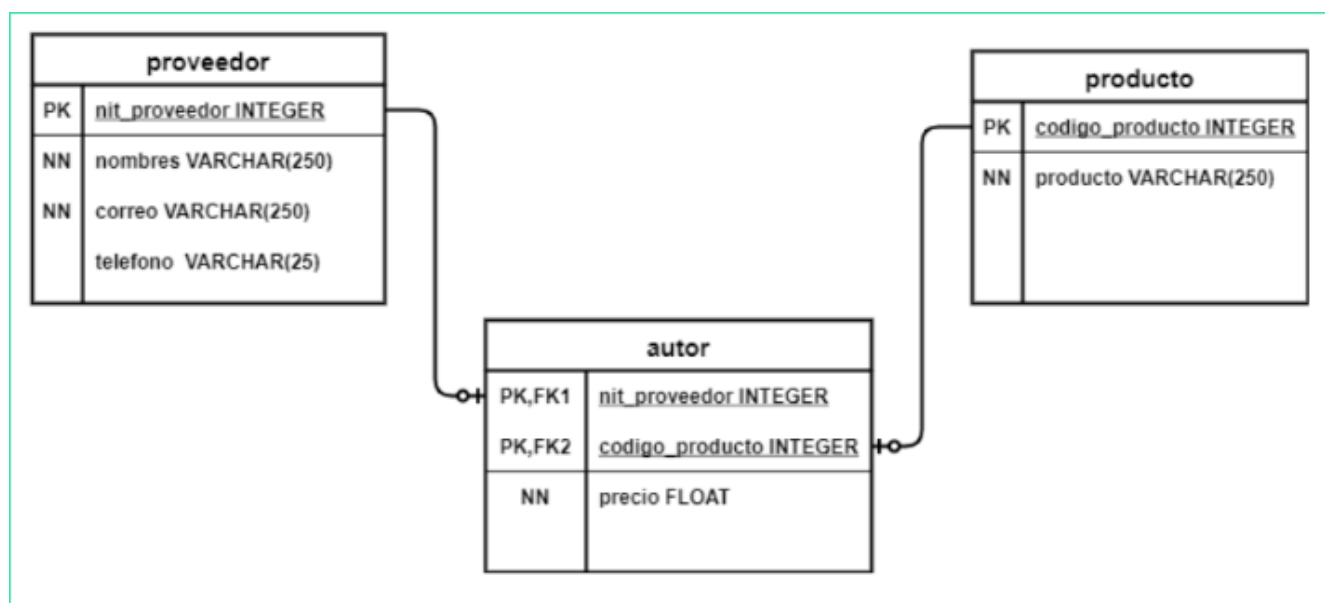
Las dependencias funcionales de la tabla 10 se pueden representar en el modelo entidad-relación, considerando que un proveedor puede suministrar varios productos y, a su vez, un producto puede ser suministrado por más de un proveedor. Además, para cada producto suministrado por un proveedor, existe un precio específico.

Figura 22. Diagrama entidad relación dependencia funcional



Como se puede notar en la figura 22, la relación provee, cuando es mapeado a un diagrama relacional, se genera una tabla intermedia provee que debe tener un atributo adicional “precio”.

Figura 23. Diagrama relacional dependencia funcional



3.2. Diseño relacional

Hasta este punto se han examinado problemas concretos de las formas normales muy básicas y de la normalización. Ahora se estudiará el modo en que se encaja la normalización en proceso general de diseño de bases de datos. Se puede hacer de una de esas formas:

- Convirtiendo un diagrama entidad relación a un diagrama relacional y de esa forma obtener las tablas.
- Puede tener una tabla con todos los datos que se procesan en una aplicación, e identificando las dependencias funcionales de manera que nos permite identificar las tablas y las llaves o claves primarias de cada una.

- c) Se puede hacer un diseño ad hoc (para que cumpla el requerimiento) y se debe comprobar luego que satisface la forma normal deseada.

En el mundo práctico y en la medida que se tiene más experticia, las opciones b y c son las más empleadas.

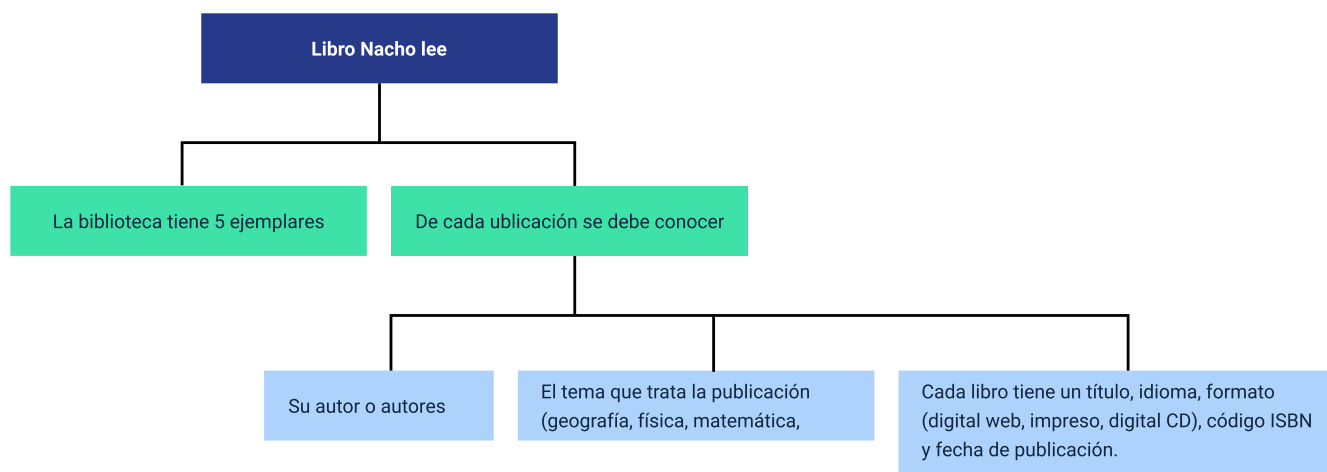
A continuación, se muestra un ejemplo completo de cómo abordar un problema de almacenamiento de datos y transformarlo en un modelo relacional, recogiendo los conceptos vistos hasta ahora, pasando de un diagrama entidad relación a un modelo relacional que es en todo caso el objetivo general de un diseño de base de datos.

Problema:

Para gestionar una biblioteca escolar, nuestro cliente requiere herramientas específicas para controlar publicaciones como libros, revistas y folletos, así como para llevar un registro de los estudiantes y los préstamos. Es necesario controlar los ejemplares de cada libro o publicación, incluyendo su ubicación y estado, y registrar tanto la retirada de libros por préstamo como la restitución de los ejemplares. Además, se necesita información sobre las editoriales, incluyendo datos como la dirección, el teléfono y el nombre de la editorial.

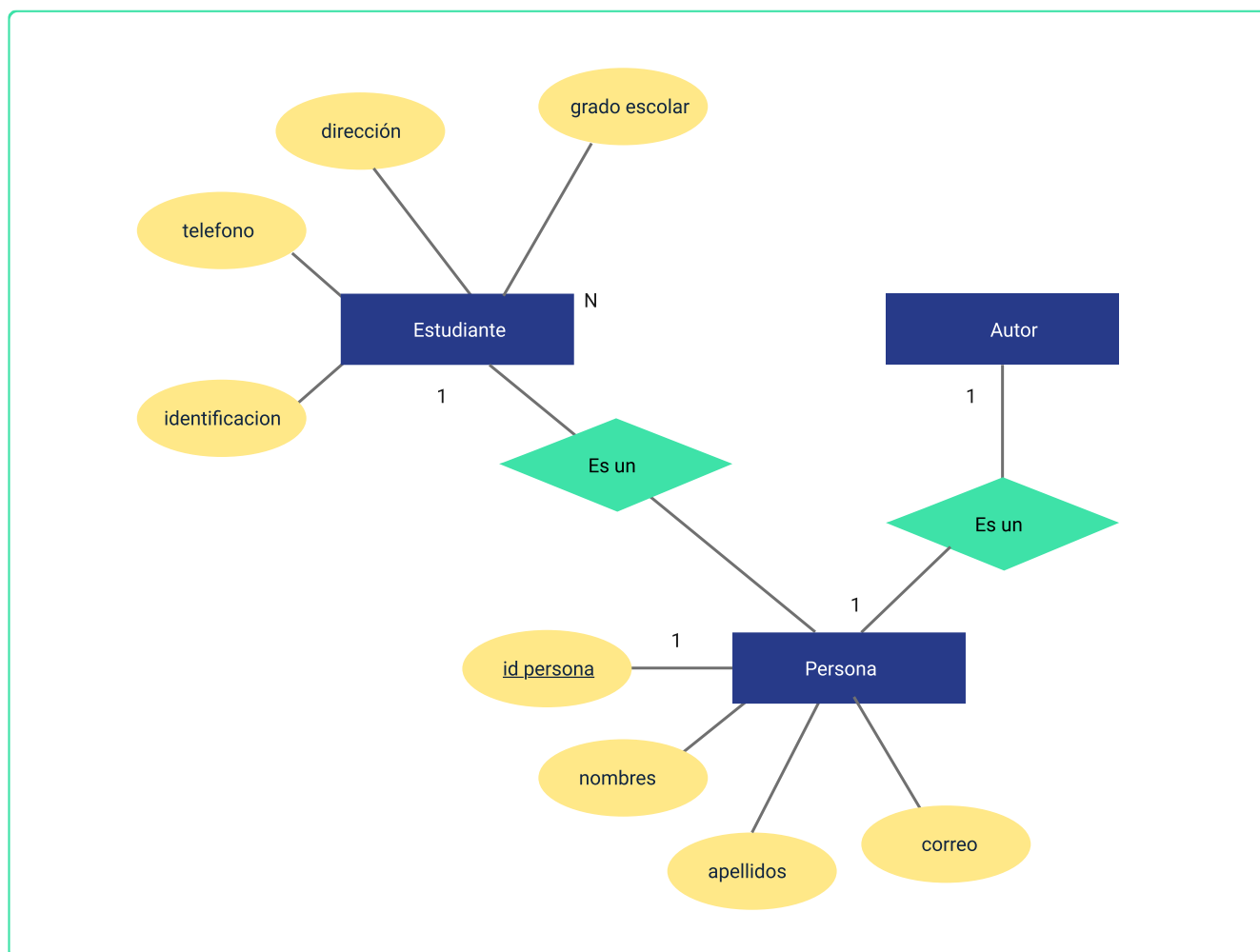
De cada publicación se pueden tener varios ejemplares, por ejemplo:

Figura 24. Diagrama de flujo "Libro Nacho lee"



- Cada ejemplar tiene un código de barras, cuando un ejemplar ya está muy descargado, se debe poder dar de baja, es decir, que ya se puede botar y no usarlo más.
- Tanto el estudiante, como el autor de un libro tienen los siguientes datos: nombres, apellidos, correo. Adicional a un estudiante tiene los datos de documento de identidad, teléfono, dirección, grado escolar.
- De cada préstamo interesa saber qué ejemplar se prestó, fecha en la que se prestó, la fecha en que se devolvió y alguna nota o comentario.

Figura 25. Análisis de relacionar autor



En este diagrama se presenta una posible solución:

La relación entre persona y estudiante corresponde al caso estudiado en las relaciones de uno a uno (entre persona y empleado del banco).

El tema se vuelve complejo porque se debe analizar si aplica lo mismo para el autor:

Al mapear estas entidades a diagrama relacional resultan 3 tablas (estudiante, persona y autor), pero autor solo tendría una única columna y al no tener atributos

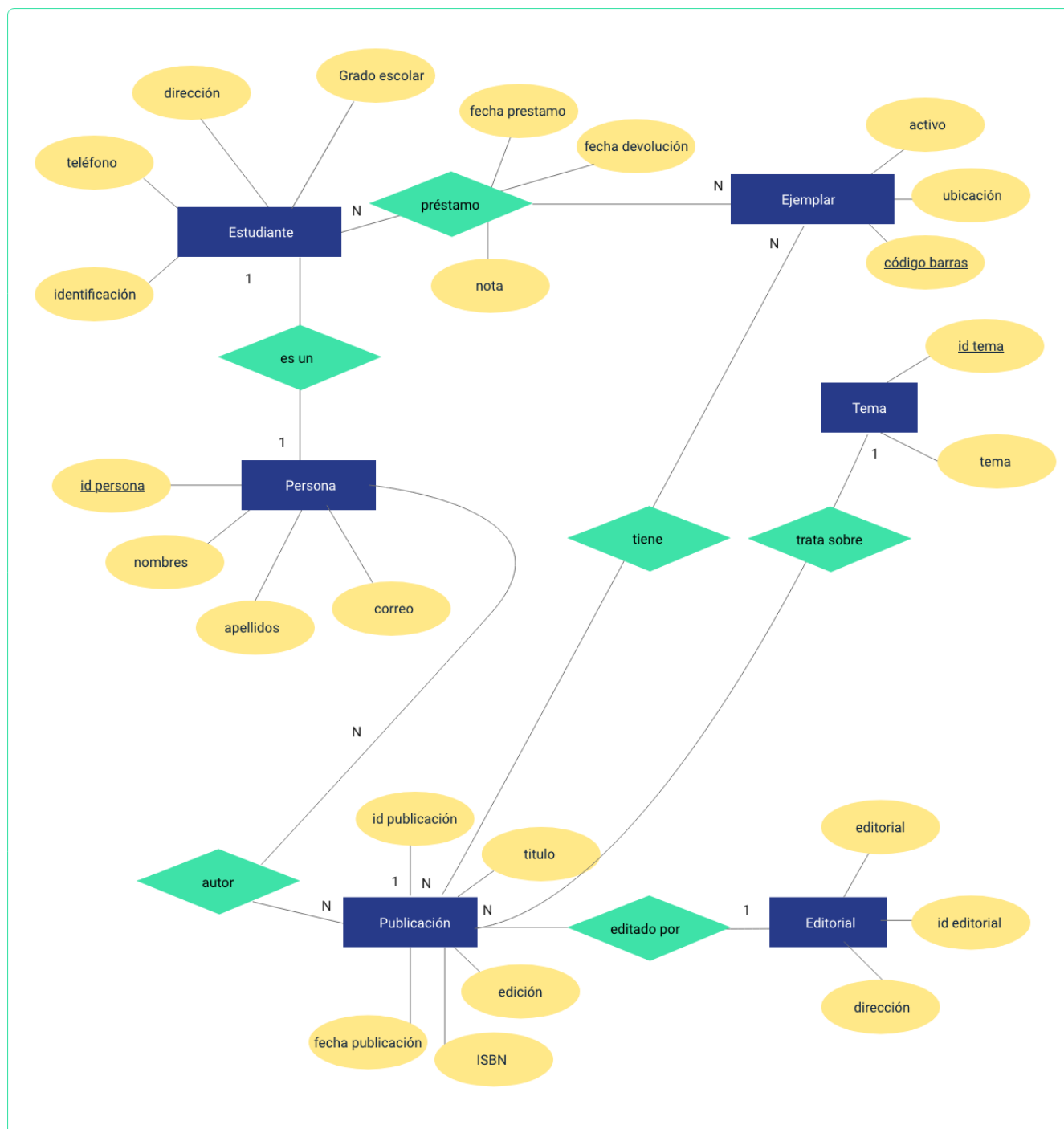
adicionales relacionados con el autor es prácticamente lo mismo que tener la tabla persona, por lo tanto, no es conveniente dejarlo de esta forma y más bien relacionar la publicación directamente con la entidad persona.

Se debe estudiar detenidamente el diagrama entidad relación, por varios minutos, asegúrese que comprende el porqué de cada elemento y su multiplicidad. Antes de continuar, realice a mano alzada diagrama del modelo relacional:

Para recordar:

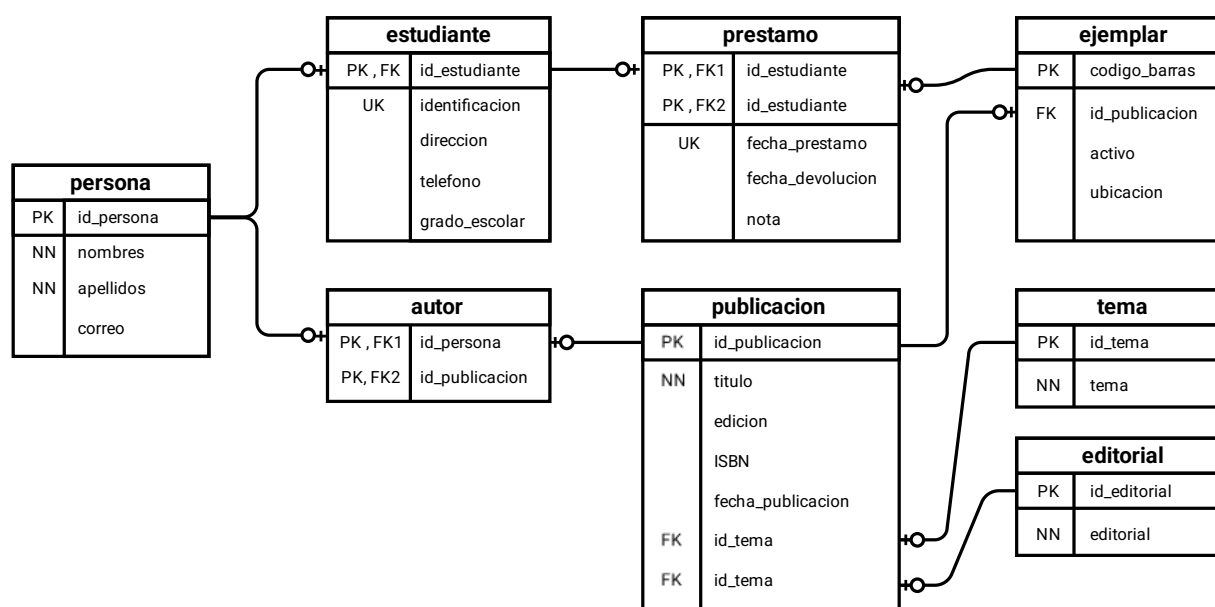
Una relación de muchos a muchos (N:N) se convierte en una tabla cuya llave primaria es computar por las llaves foráneas de las otras dos tablas, Una relación de uno a muchos (1:N) se convierte en una columna (llave foránea) de la tabla que tiene los muchos, y finalmente una relación de uno a uno (1:1) se convierte una columna en cualquier (llave foránea) de las dos tablas.

Figura 26. Modelo entidad relación biblioteca



En la siguiente figura se presenta el modelo en un diagrama relacional, este es importante, ya que hace parte de uno de los subproductos que, por petición del cliente, muchas veces es un entregable en un proyecto de “software”, no es así con el diagrama entidad relación, ya que este tipo de diagrama requiere más formas y contiene menos tecnicismos.

Figura 27. Modelo relacional



Hasta ahora se ha visto cómo distribuir el almacenamiento de los datos en tablas, de manera que no se permita la duplicidad de datos y procurando una estructura lógica, que presente la información del mundo real. Pero aún falta analizar el problema desde el punto de vista que permita garantizar que los datos que se van a insertar son correctos. Para lo cual es importante conocer las restricciones de integridad que puede tener una base de datos.

3.3. Reglas de integridad

Cuando se define un atributo, sin saber, se está escogiendo para él un dominio es decir un conjunto de valores que puede tener. Por ejemplo: en MySQL cuando se elige un atributo de tipo entero (INT) los posibles valores que puede tener esa columna van desde -2147483648 hasta 2147483647, es decir valores positivos o negativos, de manera que si se quieren sólo valores positivos se deben definirlos como entero sin signo (UNSIGNED INT), por lo tanto, los valores van desde 0 hasta 4294967295; en el siguiente enlace se puede conocer un dato completo:

- **Tipos de enteros**

En el siguiente enlace se presentan los tipos de datos enteros:

<https://dev.mysql.com/doc/refman/8.0/en/integer-types.html>

La calidad de los datos garantiza que la información almacenada en la base de datos cumpla con los estándares y requisitos de la organización, asegurando así el mantenimiento de la integridad de los datos. Esto implica aplicar un conjunto de reglas a un conjunto de datos, ya sea completo o específico, antes de almacenarlos en la base de datos de destino. Por lo tanto, para asegurar la integridad de los datos, se deben tener en cuenta los siguientes aspectos:

- **Criterio de nulidad**

Cuando en una fila un atributo (columna) es desconocido, se dice que es nulo. Un nulo no representa el valor cero (0) ni una cadena vacía (""), ya que éstos son valores tienen un significado o valor. El nulo significa ausencia de información, bien porque al insertar la fila se desconocía el valor del atributo, o tal vez porque para dicha fila el atributo no tiene valor. Debido a que los nulos no son valores, deben tratarse de forma particular,

lo que causa problemas de implementación. Actualmente, casi todos los SGDB soportan valores nulos.

- **Integridad de entidad**

Ninguno de los atributos o columnas que componen una llave primaria debe ser NULO. Es decir, una clave primaria es irreducible para identificar de modo único una fila, el que sea irreducible significa que ningún subconjunto de la clave primaria sirve para identificar inequívocamente la fila: a la fila o tupla. Si se permitiera valor nulo en una columna que hace parte de la llave primaria, esta estaría contradiciendo la irreductibilidad de una llave primaria; esto no aplica para claves alternativas, solo para la clave primaria.

- **Integridad referencial**

Para esto se debe analizar el modelo de la figura 27, y en particular, la relación entre las tablas persona y estudiante. Como se sabe, la tabla estudiante tiene una llave primaria que, a la vez, es una referencia de la tabla persona, en la siguiente figura se observa un ejemplo:

Figura 28. Integridad referencial

id_persona	correo	nombres	apellidos	id_estudiante	identificacion	direccion	telefono	grado_escolar
1	ana@gmail.com	Ana	Mendez	3	12.345.565	carrera 3 # 18- 45	7825992	6
2	jose@gmail.com	Jose Omar	Ciano	5	12.490.345	Carrera 8a No. 7 - 3	3305040	3
3	luis@gmail.com	Luis Dario	Marquez	1	13.767.234	Calle 15 No. 9 - 56	7273995	5
4	arm@gmail.com	Armando	Bronca Segura	7	13.498.543	Carrera 7A No. 32	8723041	7
5	victor@gmail.com	Vistor José	Diento Paz	8	13.242.343	Calle 12 No. 4 - 19	4210220	8
6	josen@gmail.com	Jose Nicolás	Timado Ruiz	6	19.833.574	Calle 7 No. 5 - 25	6733643	7
7	m	Andrea	Martinez					

a) Tabla persona

id_estudiante	identificacion	direccion	telefono	grado_escolar
3	12.345.565	carrera 3 # 18- 45	7825992	6
5	12.490.345	Carrera 8a No. 7 - 3	3305040	3
1	13.767.234	Calle 15 No. 9 - 56	7273995	5
7	13.498.543	Carrera 7A No. 32	8723041	7
8	13.242.343	Calle 12 No. 4 - 19	4210220	8
6	19.833.574	Calle 7 No. 5 - 25	6733643	7

b) Tabla estudiante

En la figura 27, por cada estudiante (tabla estudiante) existe uno y solo un elemento en la tabla persona, de tal forma que un estudiante es la conjunción de ambas tablas por medio de la llave foránea (que a la vez es llave primaria) `id_estudiante`, pero ¿qué tal que no existiera? Eso significa que la base de datos es inconsistente o que tiene datos basura en la tabla estudiante. Afortunadamente los SGDB permiten definir reglas que impidan que esto ocurra en las tablas, es decir que no permita registrar un estudiante con `id_estudiante` con un valor que no existe en la columna `id_persona` de la tabla persona.

Pregunta de análisis de datos

¿Qué pasaría si se borra el registro de la tabla persona que tiene `id_persona` con valor tres (3) sufrirá algún problema los datos de la tabla estudiante?, o que pasaría si se cambia el valor de tres (3) en la columna `id_persona`, por otro valor como por ejemplo nueve (9), antes de continuar, hay que tomarse unos minutos para pensarlo.

Respuesta

La respuesta a las cuestiones planeadas es que en cualquiera de las operaciones de borrado o actualización del dato se incurrirá en una violación de la integridad referencial de la tabla estudiante, ya que no habrá manera de conocer los datos de complementarios del estudiante que se encuentran en la tabla persona.

Para evitar estos problemas de inconsistencia, los sistemas gestores de bases de datos permiten definir unas reglas cuando se borra o actualiza un dato que es referencia foránea de otra tabla.

Regla de borrado: define el comportamiento a la pregunta: ¿qué ocurre si se intenta borrar la fila referenciada por la clave foránea de otra tabla? Para lo cual el SGBD implementa una de las siguientes opciones:

- **Restringir:**

No se permite borrar la fila referenciada.

- **Cascada:**

Se borra la fila referenciada y también se borran las filas que la referencian mediante la clave foránea en la otra tabla.

- **Poner “null”:**

Se borra la fila referenciada y las filas que la referenciaban ponen en nulo la clave foránea (solo si acepta nulos).

- **Valor por defecto:**

Se borra la fila referenciada y las filas que la referenciaban ponen en la clave foránea el valor por defecto definida la regla.

Regla de edición: define el comportamiento a la pregunta: ¿qué sucede si se intenta modificar el valor de la clave primaria de la fila referenciada por la clave foránea por otra tabla?

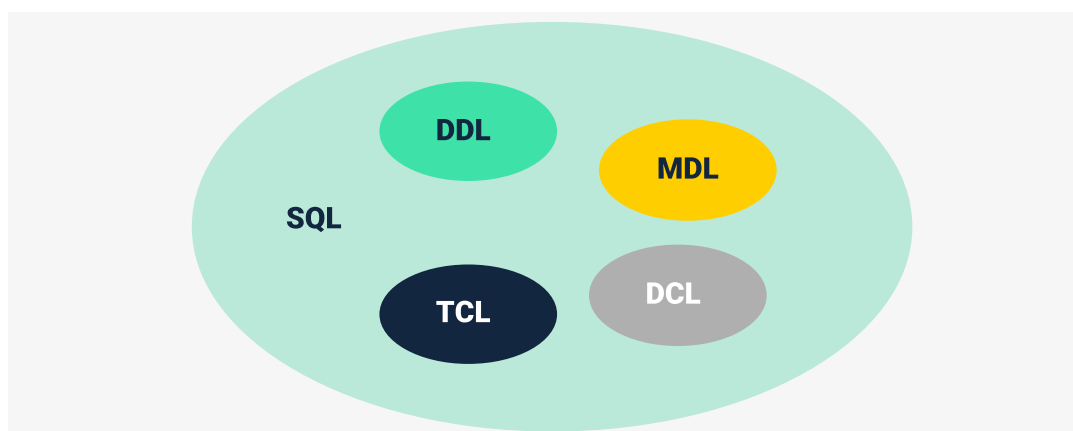
De esta forma, una vez definida la estructura del modelo de datos, se debe determinar el comportamiento de estas operaciones, lo cual generalmente responde al requerimiento del sistema de información.

3.4. Lenguajes de los sistemas administradores de bases de datos

Los sistemas gestores de bases de datos emplean un lenguaje que se denomina SQL que corresponde al nombre en inglés (“Structured Query Language”),

Generalmente, cuando un SGBD relacional implementa el lenguaje SQL, todas las acciones, y operaciones se llevan a cabo en el sistema mediante sentencias de este lenguaje. Dentro de SQL hay varios tipos de sentencias que se agrupan en cuatro conjuntos.

Figura 29. Conjunto bases de datos SQL



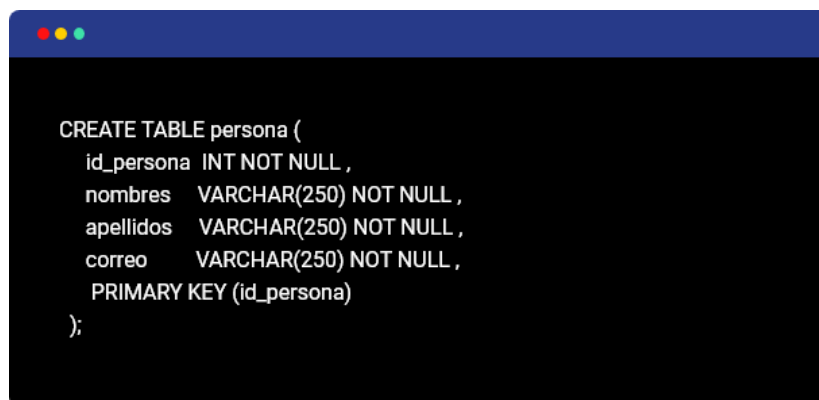
Sentencias de definición de datos (DDL):

Son las sentencias que permiten crear tablas, alterar su definición y eliminarlas. En una base de datos relacional existen otros tipos de objetos además de las tablas, como las vistas, los índices y los disparadores. Las sentencias para crear, alterar y eliminar vistas e índices también pertenecen a este conjunto de sentencia, se les denomina DDL del inglés “Data Definition Language”.

Ejemplo DDL

Para crear la tabla persona de la figura 27, quedaría de la siguiente forma:

Figura 30. Crear tabla persona



```
CREATE TABLE persona (  
    id_persona INT NOT NULL ,  
    nombres VARCHAR(250) NOT NULL ,  
    apellidos VARCHAR(250) NOT NULL ,  
    correo VARCHAR(250) NOT NULL ,  
    PRIMARY KEY (id_persona)  
);
```

Nota: cada atributo se define en una línea separado por una coma, se agregan restricciones de no nulidad (“NOT NULL”) al final la sentencia termina con punto y coma. Ahora, así se borraría esa misma tabla:

DROP TABLE persona;

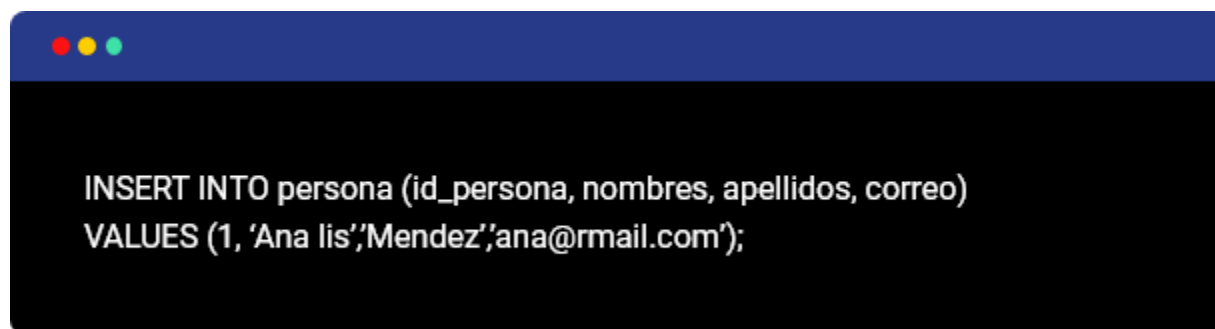
Sentencias de manejo de datos (DML):

Estas sentencias que permiten insertar datos en las tablas, consultarlos, editarlos y borrarlos, se denomina DML del inglés “Data Manipulation Language”.

Ejemplo DML

A continuación, se muestra cómo se inserta una fila a la tabla persona, luego cómo se actualiza la fila insertada y posteriormente cómo se borra:

Figura 31. Insertar una fila a la tabla persona



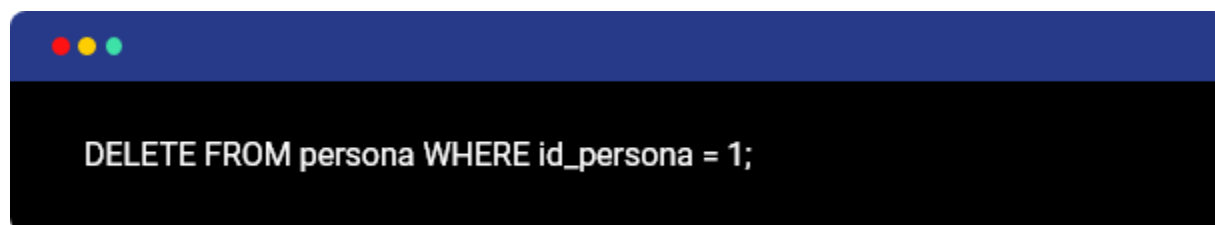
```
INSERT INTO persona (id_persona, nombres, apellidos, correo)
VALUES (1, 'Ana lis','Mendez','ana@rmail.com');
```

Figura 32. Actualizar fila insertada



```
UPDATE persona SET nombres = 'Ana Lis' WHERE id_persona = 1;
```

Figura 33. Borrar fila insertada



```
DELETE FROM persona WHERE id_persona = 1;
```

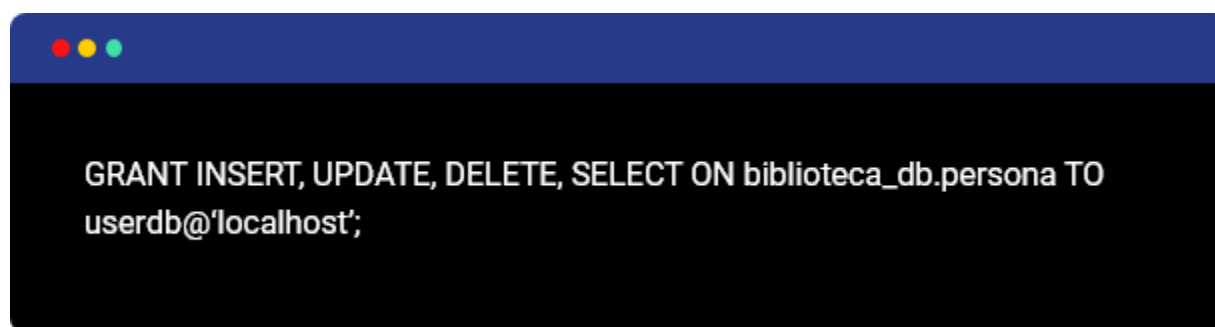
Sentencias de control (DCL):

Son las sentencias empleadas por los administradores de la base de datos para realizar tareas como, por ejemplo, crear usuarios y concederles o revocar los privilegios. Se usa el término DCL del inglés “Data Control Language”.

Ejemplo DCL

En el ejemplo se le da permisos a un usuario llamado userdb de la base de datos de nombre biblioteca_db para que pueda insertar, actualizar, borrar y consultar datos de la tabla persona desde el mismo computador donde está instalada la base de datos (“localhost”).

Figura 34. Sentencias de control



Sentencias de control de transacciones (TCL):

Un pequeño grupo de sentencias que permiten procesar en bloque operaciones DML garantizando que se efectúen todas y cada una de las operaciones o ninguna. TCL del inglés “Transaction Control Language”.

4. Sistema gestor de base de datos

En el mercado existen muchos sistemas gestores de bases de datos, a continuación, se usará MySQL para realizar las prácticas, pero ello no significa ni que sea el mejor, o el más completo del mercado, sino más bien el más usado para aplicaciones web, de pequeña, mediana y, en algunos casos, de gran complejidad. Es un buen punto de partida y todos los conceptos en MySQL vistos son reutilizables en PostgreSQL, Oracle, u otro motor de bases de datos relacional.

A continuación, se podrá conocer el paso a paso de la descarga e instalación del gestor MySQL “Workbench” y Web XAMPP:

Video 1. Descarga e instalación de herramienta CASE MySQL “Workbench”



[Enlace de reproducción del video](#)

Síntesis del video: Descarga e instalación de herramienta CASE MySQL “Workbench”

Video tutorial del experto donde se dan las indicaciones para instalar MySQL “Workbench” desde la descarga para las diferentes opciones de sistemas operativos, haciendo el registro para crear la cuenta de Oracle, necesario para hacer la descarga, y todo el paso a paso para la instalación completa, así como la ejecución de la herramienta.

Video 2. Descarga e instalación de entorno desarrollo WEB XAMPP



[Enlace de reproducción del video](#)

Síntesis del video: Descarga e instalación de entorno desarrollo WEB XAMPP

Video tutorial del experto donde se da instrucción para hacer la descarga e instalación de entorno desarrollo WEB XAMPP, el cual es un paquete de “software” que integra herramientas para el desarrollo de aplicaciones web.

Finalmente, se debe revisar cuidadosamente cómo usar MySQL “Workbench” para diseñar la base de datos del Modelo relacional de la biblioteca del numeral 3.2:

Video 3. Diseño de una base de datos con -MySQL “Workbench” relación de uno a uno



[Enlace de reproducción del video](#)

Síntesis del video: Diseño de una base de datos con -MySQL “Workbench” relación de uno a uno

Video tutorial del experto donde se da introducción para crear un modelo relacional con la herramienta “Workbench”. Verificar que esté funcionando el sistema gestor de base de datos MySQL “Workbench” y una vez abierta la herramienta MySQL “Workbench” se encuentran las indicaciones para crear un modelo.

Video 4. Diseño de una base de datos con MySQL “Workbench” relación de uno a muchos



[Enlace de reproducción del video](#)

Síntesis del video: Diseño de una base de datos con relación de uno a muchos

Video tutorial del experto donde se dan las indicaciones para crear de manera funcional una relación de uno a muchos, explorando todas las herramientas necesarias para hacer este ejercicio.

Video 5. Diseño de una base de datos con MySQL “Workbench” relación de muchos a muchos



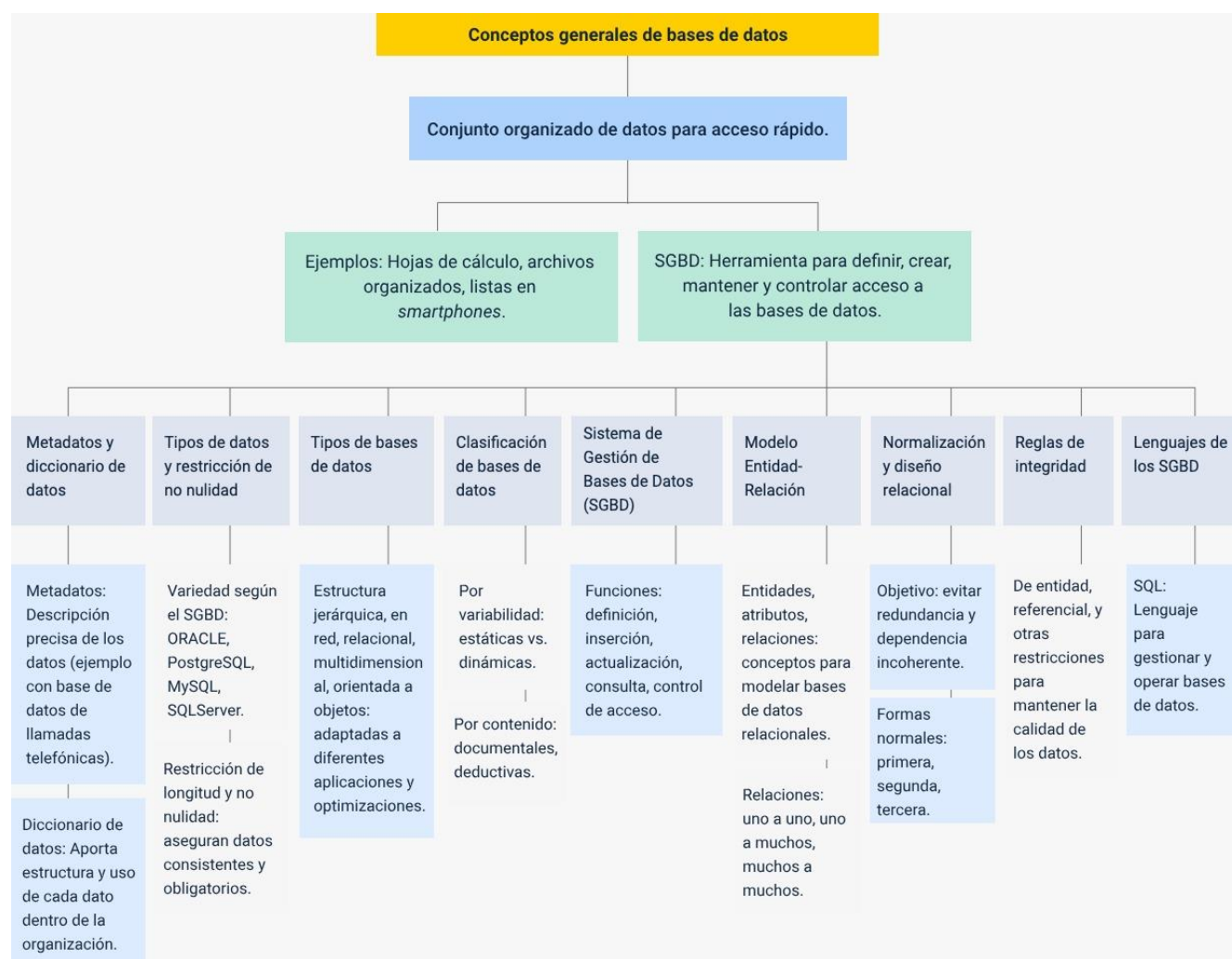
[Enlace de reproducción del video](#)

<p>Síntesis del video: Diseño de una base de datos con MySQL “Workbench” relación de muchos a muchos</p>

<p>Video tutorial del experto donde se crea de manera totalmente práctica una relación de muchos a muchos.</p>
--

Síntesis

A continuación, se presenta una síntesis de la temática estudiada en el componente formativo.



Material complementario

Tema	Referencia	Tipo de material	Enlace del recurso
Conceptos generales de bases de datos	Espinosa, A. P. 2014. L6 1 Nociones y modelos de bases de datos. [Video]	Video	https://www.youtube.com/watch?v=pPATLxijDfw

Glosario

Base datos: conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Diccionario de datos: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Metadatos: conjunto unitario de instrucciones que permite a una computadora realizar funciones diversas, como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos.

Referencias bibliográficas

Asale y RAE. (2021). Diccionario de la lengua española.

<https://dle.rae.es/base#CiiOsQO>

Kyocera. (2021). Conceptos sobre base de datos orientada a objetos. Kyocera.

<https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/paperless/conceptos-sobre-base-de-datos-orientada-a-objetos.html#:~:text=Una%20base%20de%20datos%20orientada%20a%20objetos%20es%20un%20sistema,en%20programaci%C3%B3n%20orient>

López, J. C. (2009). Algoritmos y programación (guía para docentes). Eduteka

<http://www.eduteka.org/pdfdir/AlgoritmosProgramacion.pdf>

Microsoft. (2021). Descripción de normalización de base de datos. Microsoft.

<https://docs.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description>

Resnick, M. (2007). Sembrando las semillas para una sociedad más creativa.

Laboratorio de medios de MIT, Massachussets. Eduteka.

<http://www.eduteka.org/ScratchResnickCreatividad.php>

Silberschatz, A., Korth, H. F., Sudarshan, S., Pérez, F. S., Santiago, A. I., y Sánchez, A. V. (2006). Fundamentos de bases de datos. McGrawHill.

Créditos

Nombre	Cargo	Centro de Formación y Regional
Milady Tatiana Villamil Castellanos	Responsable del Ecosistema	Dirección General
Olga Constanza Bermudez Jaimes	Responsable de Línea de Producción	Centro de Servicios de Salud - Regional Antioquia
Henry Eduardo Bastidas Paruma	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Peter Emerson Pinchao Solís	Experto Temático	Centro de Teleinformática y Producción Industrial - Regional Cauca
Paola Alexandra Moya	Evaluadora Instruccional	Centro de Servicios de Salud - Regional Antioquia
Carlos Julián Ramírez Benítez	Diseñador de Contenidos Digitales	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Desarrollador Fullstack	Centro de Servicios de Salud - Regional Antioquia
Edgar Mauricio Cortés García	Actividad Didáctica	Centro de Servicios de Salud - Regional Antioquia
Laura Gisselle Murcia Pardo	Animador y Productor Multimedia	Centro de Servicios de Salud - Regional Antioquia
Wilson Andrés Arenales Cáceres	Locución	Centro de Servicios de Salud - Regional Antioquia
María Carolina Tamayo López	Locución	Centro de Servicios de Salud - Regional Antioquia
German Acosta Ramos	Locución	Centro de Servicios de Salud - Regional Antioquia

Nombre	Cargo	Centro de Formación y Regional
Luis Gabriel Urueta Álvarez	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Jaime Hernán Tejada Llano	Validador de Recursos Educativos Digitales	Centro de Servicios de Salud - Regional Antioquia
Margarita Marcela Medrano Gómez	Evaluador para Contenidos Inclusivos y Accesibles	Centro de Servicios de Salud - Regional Antioquia
Daniel Ricardo Mutis Gómez	Evaluador para contenidos inclusivos y accesibles	Centro de Servicios de Salud - Regional Antioquia