**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | Programación de aplicaciones y servicios para la nube |
| --- | --- |

| COMPETENCIA |  | RESULTADOS DE APRENDIZAJE |  |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 007 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | 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. |
| PALABRAS CLAVE | Base de datos, relacional, integridad, dependencias |

| ÁREA OCUPACIONAL | no aplica |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

**Introducción**

1. **Conceptos generales de bases de datos**
   1. Tipos de datos y restricción de no nulidad
   2. Tipos de bases de datos
   3. Clasificación de bases de datos
   4. Sistema de gestión de bases de datos
2. **Modelo entidad relación**
   1. Relaciones entre entidades
   2. Relaciones de uno a muchos
   3. Relaciones de un uno a muchos
   4. Relaciones de uno a uno
3. **Normalización**
   1. Dependencias funcionales
   2. Diseño relacional
   3. Reglas de integridad
   4. Lenguajes de los sistemas administradores de bases de datos
4. **Sistema gestor de base de datos**
5. **DESARROLLO DE CONTENIDOS:**

**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).

| Ilustración de alojamiento de sitio web degradado | 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**

**Determinar los metadatos del siguiente problema.**

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 | 00:05:33 | ANSWERED |
| 2021-04-10 00:27:19.0 | 3023946875 | 666 | 00:00:43 | ANSWERED |
| 2021-04-10 00:22:52.0 | 3174377015 | 666 | 00:05:40 | ANSWERED |
| 2021-04-10 00:24:20.0 | 3208098045 | 666 | 00:08:09 | ANSWERED |
| 2021-04-10 00:29:28.0 | 3162400761 | 666 | 00:05:19 | ANSWERED |
| 2021-04-10 00:45:23.0 | 7005 | 883216361744 | 00:00:55 | ANSWERED |
| 2021-04-10 00:46:59.0 | 7005 | 883164829206 | 00:00:03 | ANSWERED |
| 2021-04-10 00:47:38.0 | 7005 | 883164829206 | 00:00:39 | ANSWERED |
| 2021-04-10 00:57:04.0 | 3143259110 | 666 | 00:05:23 | ANSWERED |
| 2021-04-10 01:29:16.0 | 7004 | 0 | 00:00:08 | ANSWERED |
| 2021-04-10 01:31:19.0 | 22198329 | 2 | 00:12:04 | ANSWERED |
| 2021-04-10 01:55:44.0 | 88362500 | 444 | 00:06:36 | ANSWERED |
| 2021-04-10 02:02:38.0 | 88362500 | 666 | 00:03:01 | ANSWERED |
| 2021-04-10 02:14:10.0 | 3173678610 | 666 | 00:01:49 | ANSWERED |
| 2021-04-10 02:55:15.0 | 3184096662 | 666 | 00:04:33 | ANSWERED |
| 2021-04-10 01:29:30.0 | 7004 | 1 | 01:35:05 | ANSWERED |
| 2021-04-10 03:04:30.0 | 7004 | 0 | 00:00:05 | ANSWERED |
| 2021-04-10 03:05:35.0 | 7005 | 883146610566 | 00:02:35 | ANSWERED |
| 2021-04-10 03:05:41.0 | 3003264734 | 666 | 00:02:56 | ANSWERED |
| 2021-04-10 03:18:00.0 | 7005 | 883206657698 | 00:00:00 | ANSWERED |
| 2021-04-10 03:18:50.0 | 7005 | 147584584 | 00:00:00 | NO ANSWER |
| 2021-04-10 03:19:06.0 | 7005 | 883206657698 | 00:00:02 | ANSWERED |
| 2021-04-10 03:19:49.0 | 7005 | 883206657698 | 00:00:00 | NO ANSWER |
| 2021-04-10 03:21:53.0 | 7005 | 883113084181 | 00:00:00 | NO ANSWER |
| 2021-04-10 03:22:40.0 | 7005 | 883113084181 | 00:00:00 | NO ANSWER |

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 digito (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 de la figura 1 es:

**calldate TIMESTAM(1),**

**src VARCHAR(25),**

**dst VARCHAR(25),**

**src TIME,**

**disposition VARCHAR(10)**

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

**Ejemplo 2**

Conociendo el objetivo de la base 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 datos base de llamadas telefónicas*

| NOMBRE | Base datos de llamadas telefónicas. |
| --- | --- |
| CREACIÓN | 27/02/2021 |
| DESCRIPCIÓN | Registro de las llamadas telefónicas de la PBX de la empresa STD LTDA. |

| **CAMPO** | **TIPO 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. |

| Icono de advertencia de notificación azul señal de sms de atención y mensaje de Internet. Representación 3D. | 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. **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, PostgeSQL, 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 | DATETIME2 |
| 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 **N**OT **N**ULL.

* 1. **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:

| **SLIDE**  **CF07\_1.2\_Tipos de bases de datos** |
| --- |

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. **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**  
*Clasificación de bases de datos*

| Criterio de clasificación | Clasificación | Descripción |
| --- | --- | --- |
| Según la variabilidad de los datos. | Bases de datos estáticas. | * 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. * Generalmente, a este tipo de bases de datos se les llama bodega de datos. * Muchas de estas bases de datos de 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*). |
| Según la variabilidad de los datos. | Bases de datos dinámicas. | * 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. * 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*). |
| 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*. **La base de datos relacionales es de obligatorio dominio en cualquiera de los casos, ya que los conceptos de esta son reutilizables en casi todos los otros tipos.**

* 1. **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:



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.

1. **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:

| **Infografía interactiva**  **CF07\_2\_Modelo entidad relación** |
| --- |

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*

| identificacion | 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 |
| 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.

* 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.



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:**  En la tabla 6, es sabido por todos que no existen dos personas con el mismo número de documento de identidad, por lo tanto, se puede emplear el atributo identificación como clave de la tabla persona. Así mismo, no deben existir dos cuentas bancarias con el mismo número de cuenta, entonces, el atributo número\_cuenta se puede emplear como clave de la tabla cuenta. Esto se puede presentar en un diagrama de entidad relación cuando a los atributos que componen una clave se les marca con subrayado el nombre del atributo tal como se muestra en la siguiente figura.  **Figura 1** *Claves de las entidades persona y cuenta*    Nótese como los atributos **identificacion** y **numero\_cuenta** tiene 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.



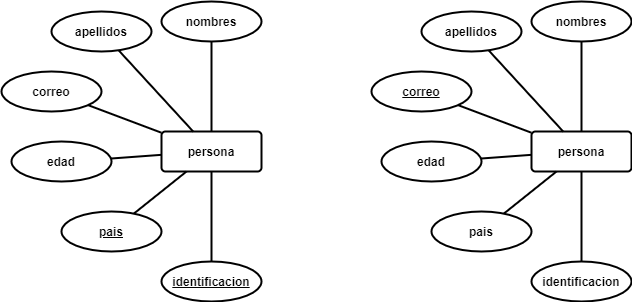
* **Clave compuesta**

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.



Estos dos casos se ilustran en la figura siguiente.

**Figura 2**  
*Claves candidatas*



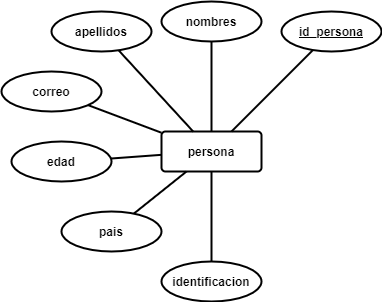
* **Clave candidata**

Los dos caminos propuestos en la figura 2 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.



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

**Figura 3**   
*Claves candidatas secuencia*



**Tabla 7**

Entidad persona

| **id\_persona** | **pais** | **identificacion** | **correo** | **nombres** | **apellidos** | **edad** |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | colombia | 8.127.346 | [ana@rmail.com](mailto:ana@rmail.com) | Ana lis | Mendez | 34 |
| **2** | colombia | 212.394.762 | [jose@pmail.com](mailto:jose@pmail.com) | Jose Omar | Ciano | 28 |
| **3** | colombia | 545.232.546 | [luis@jmail.com](mailto:luis@jmail.com) | Luis Dario | Marquez | 21 |
| **4** | colombia | 125.446.732 | [arm@hmail.com](mailto:arm@hmail.com) | Armando | Bronca Segura | 35 |
| **5** | colombia | 33.565.236.787 | [victor@umail.com](mailto:victor@umail.com) | Vistor José | Diento Paz | 24 |
| **6** | colombia | 2.343.554.768 | [josen@pmail.com](mailto:josen@pmail.com) | Jose Nicolás | Timado Ruiz | 19 |
| **7** | perú | 45.456.778 | [andrea@email.com](mailto: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.

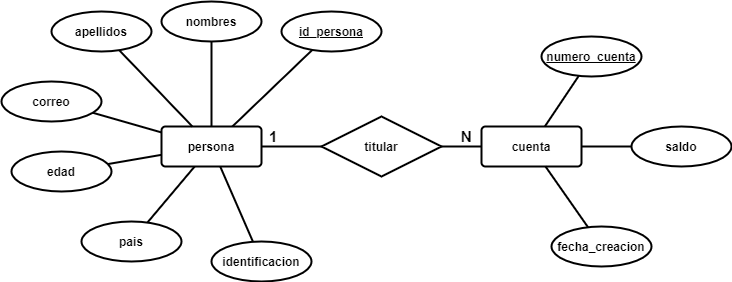
* 1. **Relaciones de uno a muchos**

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.

| **Por 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.

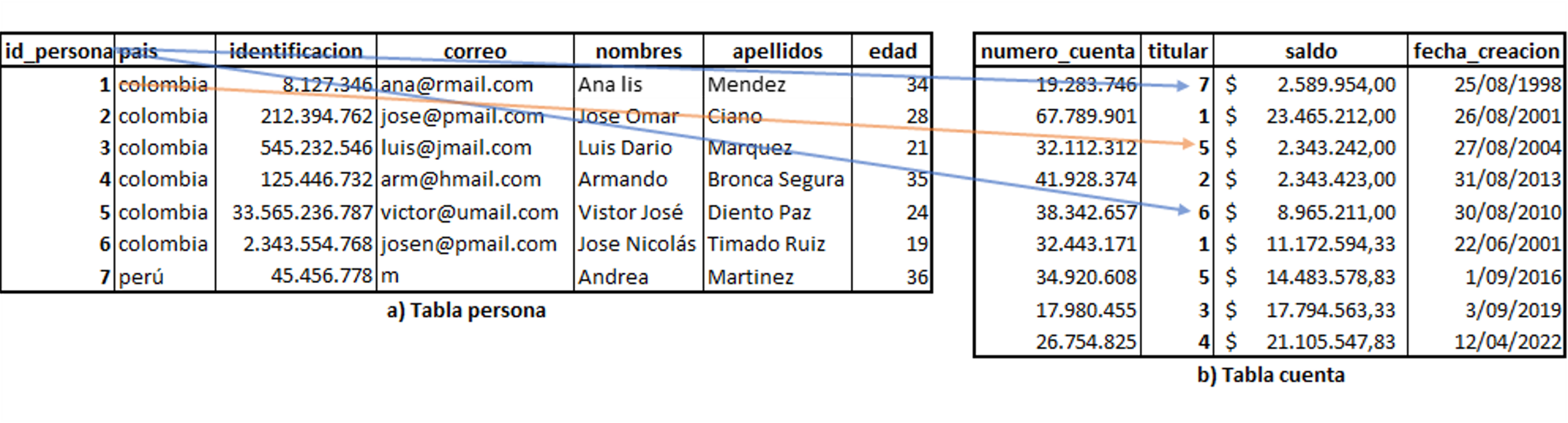
**Figura 4**  
*Relaciones de uno a muchos*



A esta representación se le denomina multiplicidad **1:N** y se lee de la siguiente forma: **UNA** **persona** es **titular** de **N** **cuenta**s 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 de es 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 5** *Tablas de un modelo relacional 1:N*

**

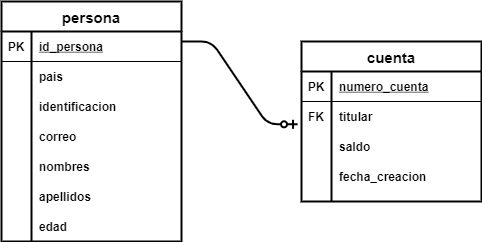
Como se evidencia en la figura 5, Ana Lis Méndez es titular de dos cuentas bancarias (67.789.901 y 32.443.171). Todos los datos en la columna titular de la **tabla cuenta** se denominan **LLAVE FORÁNEA** de la tabla persona.

**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 6**  
*Diagrama relacional 1:N*



En la figura 6 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 **P**rimary **K**ey), 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 **F**oreign **K**ey).

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

**Figura 7**  
*Diagrama relacional 1:N con metadatos*

A diagram of a number

Description automatically generated

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

* 1. **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 8** *Diagrama de entidad de publicaciones bibliográficas*    Como se identifica en la figura 8, 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.

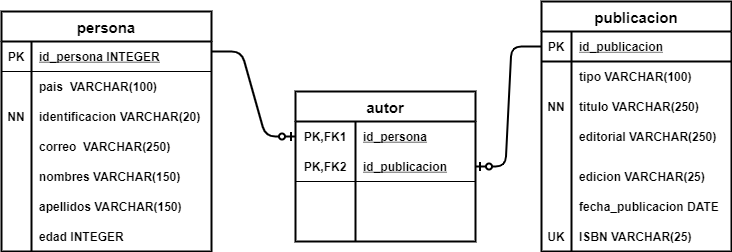
En la figura 9, la “tabla autor”, dice que la persona Ana Lis, es autora de los libros de título *Grandes hazañas* y *Pellentesque*, también que el libro *Pellentesque* es escrito además de Ana Lis por Luis Darío y Jose Nicolás. Lo anterior haciendo referencia a las llaves primarias de las tablas a) **persona** y b) **publicacion**.

**Figura 9**  
*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 **FK**1 y **FK**2, para indicar llave foránea uno y dos, para las entidades persona y publicación 2, respectivamente. También se puede identificar el prefijo **UK** (**U**nique **K**ey) 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 10**  
*Diagrama relacional N:N*



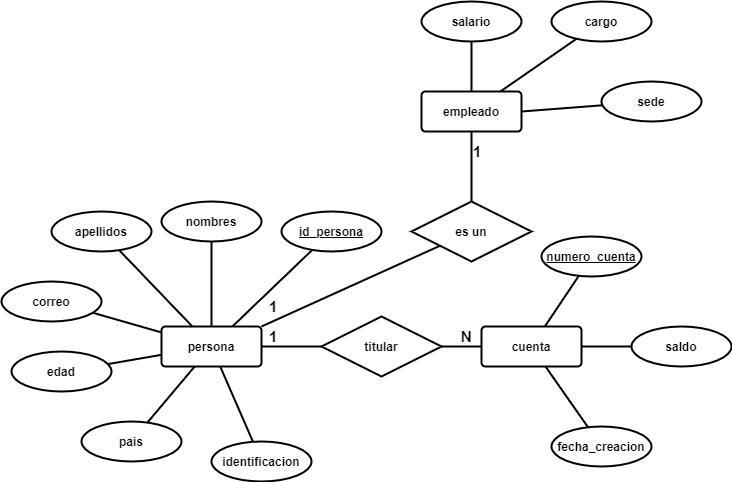
* 1. **Relaciones de uno a uno**

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:



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.

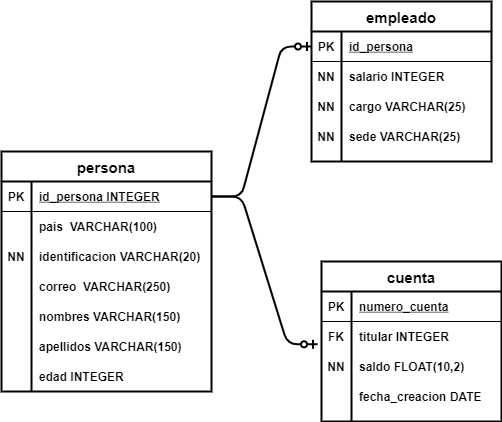
**Figura 11**   
*Diagrama entidad relación persona-empleado*



**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 (**U**nique **K**ey o **P**rimary **K**ey) como se muestra en el siguiente diagrama relacional.

**Figura 12**  
*Diagrama relacional 1:1*



Como se puede inferir de la figura 12, 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.

1. **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 12 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.



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.

| **PESTAÑAS**  **CF07\_3\_Normalización** |
| --- |

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.

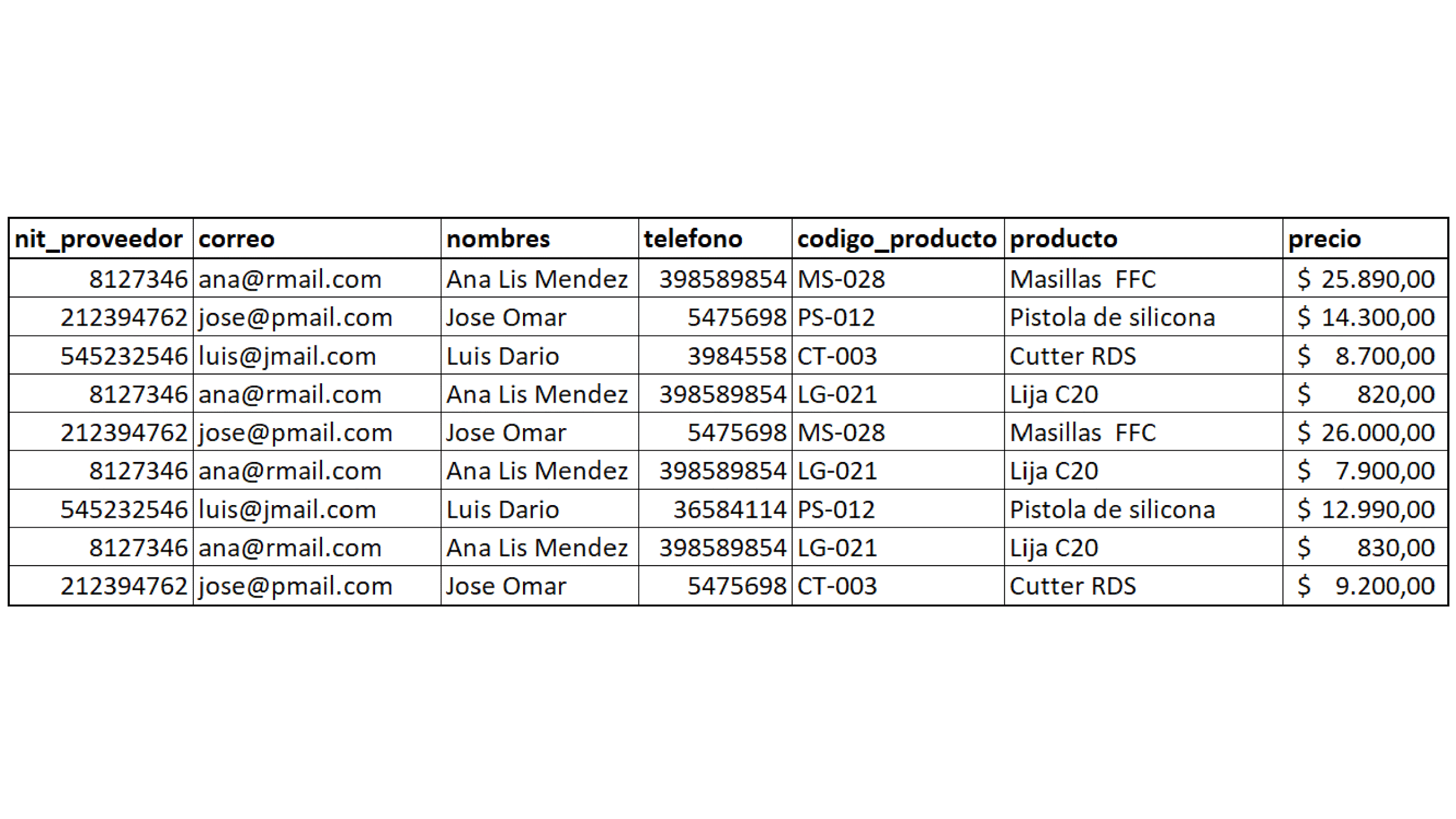
* 1. **Dependencias funcionales**

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 8** *Proveedores, productos y precios*   | **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 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

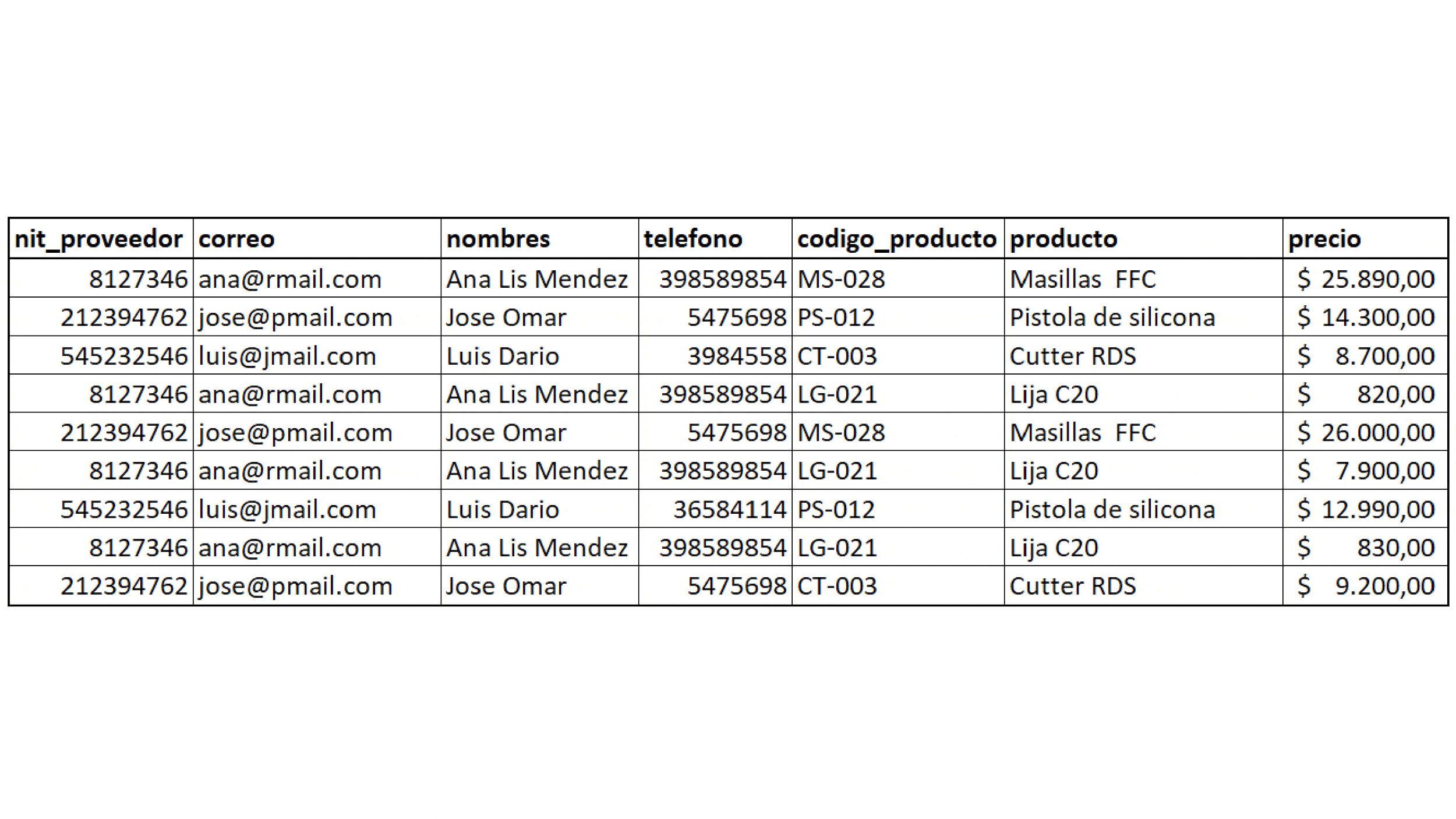
El principio de no repetición de datos sugiere que se deben identificar todos los datos que se repiten. Esto se puede conocer en la siguiente figura. Es importante destacar que los valores de **nit\_proveedor** son iguales en todos los registros que comparten el mismo valor de correo, nombres y teléfono.

**Figura 13**  
*Dependencia de nit\_proveedor con datos del proveedor*



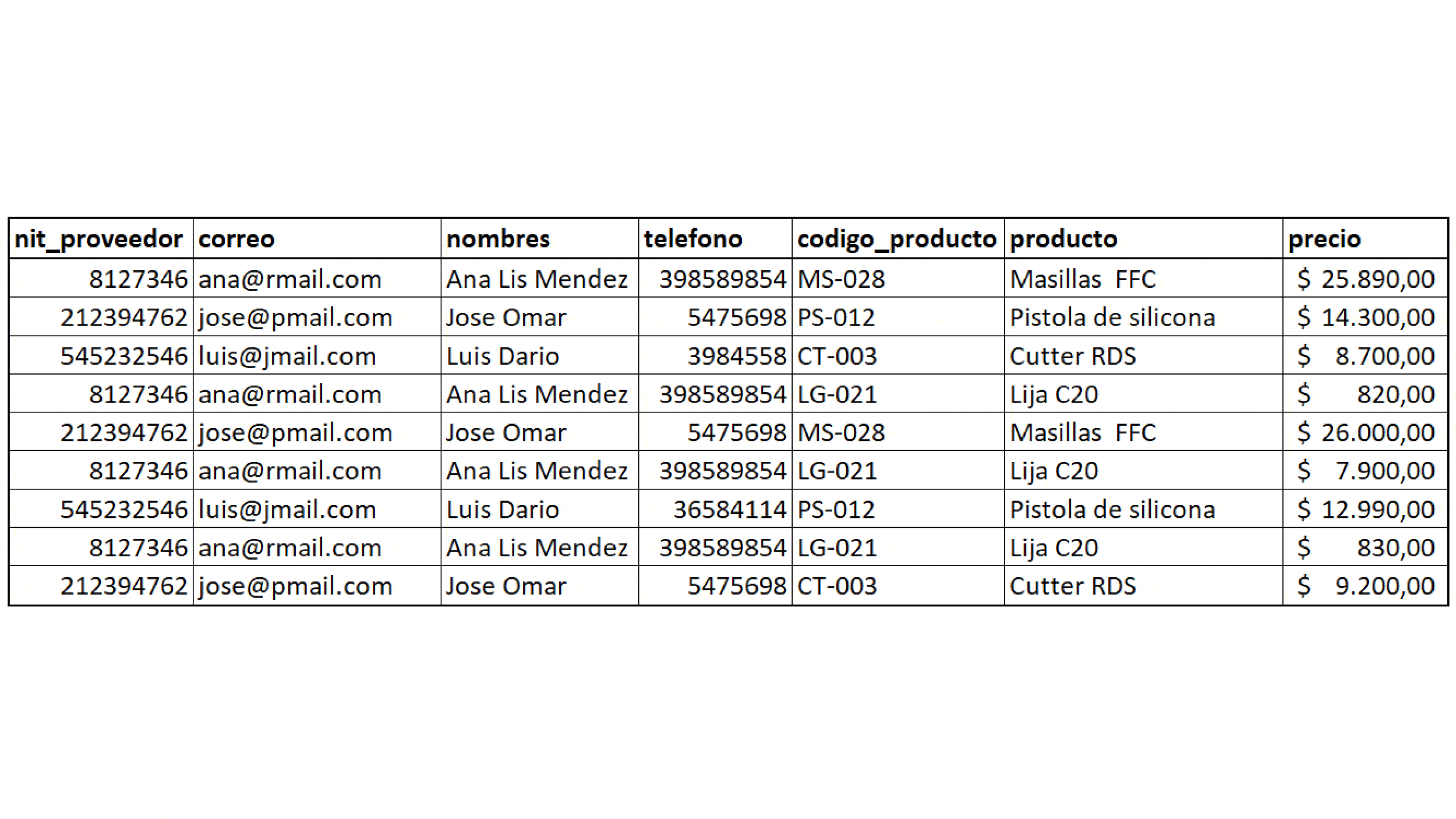
También en la figura 14 la columna **código\_producto** es la misma para la columna **producto**, pero difiere del precio que cada proveedor ofrece del mismo.

**Figura 14**  
*Dependencia de código\_producto\_ con datos del producto*

****

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 15**  
*Dependencia de nit\_proveedor, código\_producto\_ con precio del producto*



En la siguiente tabla se presentan las dependencias detectadas.

**Tabla 8**  
*Dependencias funcionales*

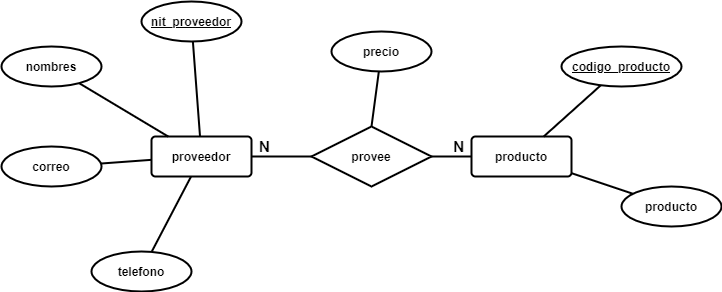
| Atributo | Es dependencia funcional de: |
| --- | --- |
| nit\_proveedor | * correo * nombres * teléfono |
| codigo\_producto | * producto |
| nit\_proveedor, codigo\_producto | * precio |

**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.

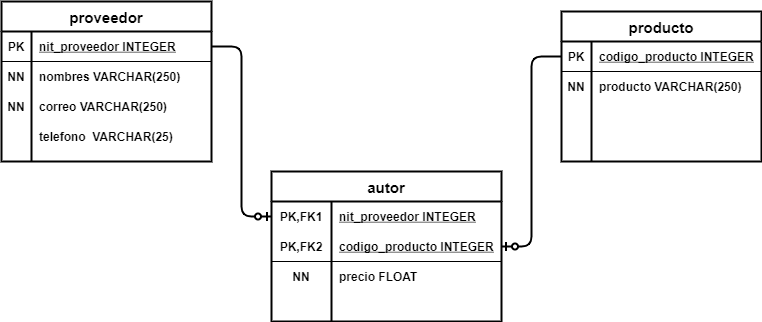
Las dependencias funcionales de la tabla 8 se pueden representar en el modelo entidad-relación, considerando que un proveedor puede suministrar varios productos y, a su vez, un producto puede ser suministrado por más de un proveedor. Además, para cada producto suministrado por un proveedor, existe un precio específico.

**Figura 16**   
*Diagrama entidad relación dependencia funcional*



Como se puede notar en la figura 16, la relación provee, cuando es mapeado a un diagrama relacional, se genera una tabla intermedia provee que debe tener un atributo adicional “precio”.

**Figura 17**  
*Diagrama relacional dependencia funcional*



* 1. **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:



En el mundo práctico y en la medida que se tiene más experticia las opciones 2 y 3 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:



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.**

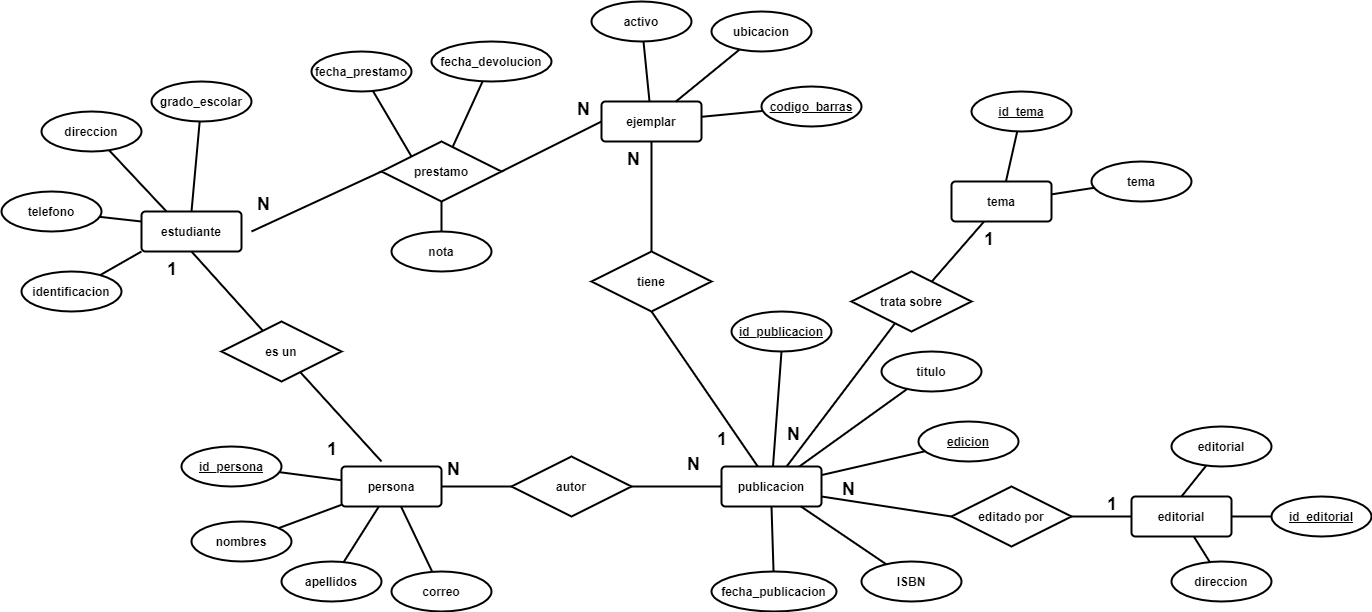
De cada préstamo interesa saber qué ejemplar se prestó, fecha en la que se prestó, la fecha en que se devolvió y alguna nota o comentario.

| Figura 18  *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 diagrame el modelo relacional:

| Bombilla de luz amarilla brillante ilustración vectorial de dibujos animados, símbolo de idea | **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 19**  
*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 20**  
*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.

* 1. **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:

| Link with solid fill | 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:

| **Acordeón**  **CF07\_3.3\_Reglas de integridad** |
| --- |

En la imagen anterior, 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.



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:



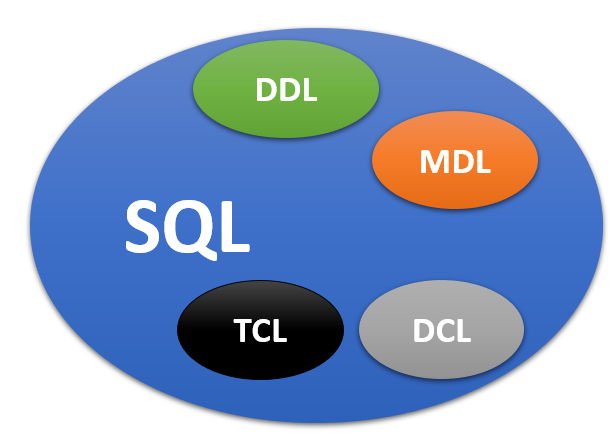
**Regla de edición:** define el comportamiento a la pregunta: ¿qué sucede si se intenta modificar el valor de la clave primaria de la fila referenciada por la clave foránea por otra tabla?

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

* 1. **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 (**S**tructured **Q**uery **L**anguage), 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 21**  
*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 **D**ata **D**efinition **L**anguage.

**Ejemplo DDL**

Para crear la tabla persona de la figura 20, quedaría de la siguiente forma:

| **CREATE TABLE** persona (  id\_persona **INT NOT NULL ,**  nombres V**ARCHAR(**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 sentía 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 **D**ata **M**anipulation **L**anguage.

**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:

| **INSERT INTO** persona (id\_persona, nombres, apellidos, correo)  **VALUES** (1, 'Ana lis','Mendez','ana@rmail.com'); |
| --- |

| **UPDATE** persona SET nombres = 'Ana Lis' WHERE id\_persona = 1; |
| --- |

| **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 **D**ata **C**ontrol **L**anguage.

**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).

| **GRANT INSERT, UPDATE, DELETE, SELECT ON** biblioteca\_db**.**persona **TO** userdb@'localhost'; |
| --- |

* **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 **T**ransaction **C**ontrol **L**anguage.

1. **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 PosgresSQL, 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 DESCARGA E INSTALACIÓN de herramienta CASE MySQL Workbench.

VIDEO DESCARGA E INSTALACIÓN de entorno desarrollo WEB XAMPP.

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:

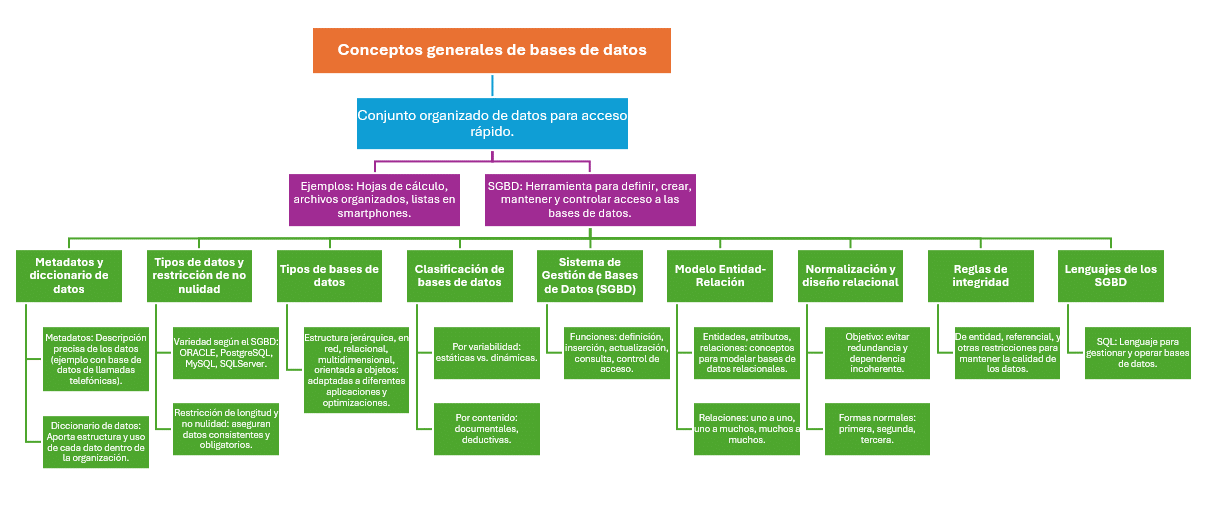
VIDEO DISEÑO DE UNA BASE DATOS con -MySQL Workbench relación de uno a uno

VIDEO DISEÑO DE UNA BASE DATOS con MySQL Workbench relación de uno a muchos

VIDEO DISEÑO DE UNA BASE DATOS con MySQL Workbench relación de muchos a muchos

1. **SÍNTESIS**

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



1. **ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la Actividad | Exploradores de datos: conectando conceptos |
| Objetivo de la actividad | Reconocer los conceptos en situaciones teóricas y prácticas, preparándolos para su uso en diversos contextos académicos y profesionales. |
| Tipo de actividad sugerida | Arrastrar y soltar |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | CF07\_Actividad didactica |

1. **MATERIAL COMPLEMENTARIO:**

| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| --- | --- | --- | --- |
| Conceptos generales de bases de datos | Espinosa, A. P. 2014*. L6 1 Nociones y modelos de bases de datos.* [Video] YouTube. <https://www.youtube.com/watch?v=pPATLxijDfw> | Vídeo | <https://www.youtube.com/watch?v=pPATLxijDfw> |

1. **GLOSARIO:**

| TERMINO | SIGNIFICADO |
| --- | --- |
| Base datos: | conjunto ordenado y finito de operaciones que permite hallar la solución de un problema. |
| 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. |
| Diccionario de datos: | conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. |

1. **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://rb.gy/5deqdh>

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.

1. **CONTROL DEL DOCUMENTO**

|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| --- | --- | --- | --- | --- |
| Autor (es) | Henry Eduardo Bastidas Paruma | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Mayo 2021 |
| Peter Emerson Pinchao Solís | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Mayo 2021 |
| Oscar Absalón Guevara | Diseñador instruccional | Centro de Gestión Industrial | Mayo 2021 |
| Rafael Neftalí Lizcano Reyes | Asesor Pedagógico Ecosistema RED | Centro Industrial del Diseño y la Manufactura - Regional Santander | Junio del 2021 |
| José Gabriel Ortiz Abella | Corrector de estilo | Centro de la Industria de la Comunicación Gráfica - Regional Distrito Capital | Junio del 2021 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| --- | --- | --- | --- | --- | --- |
| Autor (es) | Paola Alexandra Moya | Evaluadora instruccional | Centro de Servicios de Salud | Febrero 2024 | Actualización |
|  | Olga Constanza Bermúdez Jaimes | Responsable Línea de Producción Antioquia | Centro de Servicios de Salud | Febrero 2024 | Actualización |