

---

## SISTEMAS DE DATOS

---

### SISTEMAS DE BASE DE DATOS.

#### Concepto de Sistema de Base de Datos.

Conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministran, tanto a los usuarios como a los analistas, programadores o administrador los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad.

En un diseño de base de datos la aplicación no conoce como están físicamente los datos. Esta es la primera diferencia entre un sistema de base de datos y un sistema de archivos.

#### Concepto de Base de Datos.

Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más de la mejor manera posible; los datos se almacenan para incluir nuevos datos y para modificar o extraer los datos almacenados.

La base de datos es solamente la parte que corresponde a la información de un sistema de base de datos. La información se encuentra en entidades.

#### Componentes de un sistema de base de datos.

- Información o datos: Son el objeto principal y, por lo tanto, ningún sistema de base de datos puede existir sin los datos (datos interrelacionados que forman la base de datos). Se trata de información integrada y compartida, porque los diferentes usuarios acceden a los mismos datos.
- Equipamiento o hardware: Es el conjunto de dispositivos físicos sobre los que reside la BD. Consiste en una o más computadoras, unidades de disco, monitores, impresoras, unidades de almacenamiento, cableado, etc.
- Programas: un sistema de base de datos incluye varios tipos de software:
  - El *sistema de gestión* de base de datos (SGBD o DBMS).
  - El *software de aplicación*, que usa las facilidades del DBMS para manipular la BD.
  - *Herramientas* para el desarrollo de aplicaciones, como los generadores de pantallas, de menús y de informes; los generadores de aplicaciones, los compiladores y las facilidades para la definición de la vista y los datos.
- Usuarios:
  - El programador de aplicaciones.
  - El usuario final.
  - El administrador (DBA).

#### Funciones del DBMS

- Brinda las herramientas para la entrada y salida de datos (función principal).
- Se encarga del control de acceso y la seguridad.
- Actúa como nexo entre las aplicaciones y la base de datos. Las aplicaciones no actúan directamente sobre la base de datos, sino que se realiza una petición mediante el DBMS.

#### Ventajas de un sistema de base de datos.

##### Reducción de la redundancia

La redundancia es perjudicial si no está controlada. Toda redundancia que está puesta a propósito deberá estar controlada y no debe ser perjudicial. Si está controlada, existen políticas del DBA para manejarla. No existen dos entidades redundantes, pero sus atributos sí pueden ser redundantes.

Los movimientos en una base de datos se transaccionan para que exista consistencia. Una transacción puede incluir una o más acciones sobre la base de datos, y se cumplen todas o fallan todas. Esto garantiza la integridad de la base de datos.

##### Se evita la inconsistencia

N entradas a la misma petición deben obtener la misma respuesta.

Existe una inconsistencia temporal mientras se está realizando la transacción. Completado el proceso desaparece.

##### Se comparten los datos

Distintos usuarios (programas de aplicaciones o usuarios finales) comparten los mismos datos.

### Normas

Se pueden establecer normas y cumplirlas, tanto de formato como de nombre o documentación.

Todo sistema de base de datos permite implementar un diccionario de datos que informa los distintos formatos de los atributos. También sirve para documentar en qué entidad existe un atributo y qué formato tiene.

### Seguridad

Se provee seguridad tanto en los accesos como en los permisos.

### Integridad

Se mantiene la integridad garantizando que los datos son correctos. Esto es consecuencia de las ventajas anteriores.

### Independencia de los datos

Cualquier cambio que sufre la base de datos debería ser transparente para las aplicaciones y usuarios finales, debido a que la información se la da al DBMS; no es necesario realizar cambios en los programas, lo que reduce los costos de mantenimiento. Ej.: agregar un atributo.

## ARQUITECTURA DE UNA BASE DE DATOS.

El objetivo de presentar una arquitectura es establecer un marco de referencia útil para describir los conceptos generales de la base de datos. Es un modelo que trata de explicar el funcionamiento de la base de datos.

### Arquitectura de los tres niveles (interno, conceptual y externo)

#### *Nivel externo*

Es el más cercano a los usuarios. Se ocupa de la forma en que los usuarios perciben los datos.

Debido a que este esquema es la visión de la base de datos que tiene un usuario en particular, en él deberán encontrarse reflejados sólo aquellos datos e interrelaciones que necesite el usuario correspondiente.

También habrán de especificarse las restricciones de uso, como puede ser en derecho a insertar o borrar determinados datos, o el acceso a los mismos, etc.

Una **vista** es la forma en que el usuario percibe los datos. Muestra la información que el usuario necesita y en la forma en que éste la necesita. Las vistas están soportadas en los esquemas. Puede haber más de una vista sobre un esquema. Es algo lógico que no existe físicamente. No contiene ninguna definición de datos.

En los **esquemas** se encuentran las definiciones de datos (tienen un lenguaje de definición de datos). Es soportado por el nivel conceptual. Es definido por el DBA.

Habrán tantos esquemas externos como exija las diferentes aplicaciones. Un mismo esquema externo podrá ser utilizado por varias aplicaciones.

El **DSL (lenguaje de consulta)** es el lenguaje con el que trabaja la base de datos. Existe también un **lenguaje anfitrión**, que es el lenguaje en el que se programa la interfaz y que, con secuencias propias del DSL, maneja la base de datos.

En resumen, el nivel externo se ocupa de las vistas individuales de los usuarios, y puede considerarse a nivel conceptual como la vista comunitaria de todos los usuarios. Dicho de otro modo: existirán muchas vistas externas distintas, cada una formada por una representación de una parte de la base de datos total, y existirá una sola vista conceptual formada por una representación de la base de datos en su totalidad. De manera similar, existirá una sola vista interna la cual representa la base de datos tal como está almacenada internamente. En el nivel externo hay usuarios finales, aplicaciones, terminales, etc.

#### *Nivel conceptual*

Es el nivel de mediación entre los otros dos.

En este nivel debería estar la representación de la totalidad de la base de datos (toda la información que quiero procesar). Ej: UADER: nivel conceptual (toda la facultad), un esquema por institución, una vista para: profesores, bedelía, alumnos, etc.

No debe existir ninguna referencia a cómo están almacenados los datos ni a cómo están utilizados.

Este nivel está representado mediante entidades y relaciones. Cuando se habla de diseño de base de datos se habla de este nivel.

El esquema conceptual debería incluir además de la definición de datos, las verificaciones de seguridad y las verificaciones de integridad. La seguridad se verifica controlando los accesos y permisos, y la integridad, comprobando que existan las entidades y que se cumplan las relaciones.

Una vez terminado el diseño conceptual se debe pensar en el diseño físico.

### Nivel interno

Es el más cercano al almacenamiento físico. Aquí se encuentra la definición de cómo se recupera la información y de cómo está almacenada la misma.

Aunque el contenido del esquema interno depende mucho de cada SGBD, podemos distinguir tres clases de aspectos que deben especificarse en él:

- *Estrategia de almacenamiento:* En este apartado se incluye la asignación de espacios de almacenamiento para el conjunto de datos. También deberá indicarse la estrategia de emplazamiento de los datos que ha sido utilizada para optimizar tiempos de respuesta y espacio de memoria secundaria. Por último, deberán aparecer aspectos como el tratamiento de los desbordamientos, etc.
- *Caminos de acceso:* Incluiremos en los caminos de acceso las especificaciones de claves, así como de índices o punteros.
- *Miscelánea:* Además de los aspectos citados, habría que incluir, en el esquema interno, otros varios, como técnicas de compresión de datos, de criptografía, las correspondencias entre esquema interno y esquema conceptual, técnicas de ajuste o afinamiento (Tuning), optimización, etc.

### Funciones del DBA

- *Definir el esquema conceptual:* es tarea del administrador de datos decidir con exactitud cual es la información que debe mantenerse en la base de datos, es decir, identificar las entidades que interesan a la empresa y la información que debe registrarse acerca de esas entidades.
- *Definir el esquema interno:* el DBA debe decidir también como se representará la información en la base de datos almacenada. A este proceso suele llamársele diseño físico de la base de datos. Una vez hecho esto el DBA deberá crear la definición de estructura de almacenamiento correspondiente (es decir el esquema interno); además deberá definir la correspondencia pertinente entre los esquemas interno y conceptual.
- *Definir el esquema externo:* el DBA debe encargarse de la comunicación con los usuarios, garantizar la disponibilidad de los datos que requieren y escribir los esquemas externos necesarios. Además, será preciso definir la correspondencia entre cualquier esquema externo y el esquema conceptual.
- **Definir las verificaciones de seguridad e integridad:** las verificaciones de seguridad y de integridad pueden considerarse parte del esquema conceptual.
- **Definir procedimientos de respaldo y recuperación:** cuando una empresa se decide a utilizar un sistema de base de datos, se vuelve dependiente del funcionamiento correcto de ese sistema. En caso de que sufra daño cualquier porción de la base de datos resulta esencial poder reparar los datos implicados con un mínimo de retraso y afectando lo menos posible el resto del sistema. El DBA debe definir y poner en practica un plan de recuperación adecuado.
- **Supervisar el desempeño y responder a cambios en los requerimientos:** es responsabilidad del DBA organizar el sistema de modo que se obtenga el desempeño que sea “mejor para la empresa”, y realizar los ajustes apropiados cuando cambien los requerimientos.

## MODELO CONCEPTUAL CON MODELO ENTIDAD-RELACIÓN (MER).

El diseño como actividad se puede entender en distintos niveles de abstracción, separándolo en diseño conceptual, diseño lógico y diseño físico.

El diseño conceptual es el que no se encuentra asociado con ninguna plataforma de implementación, sino que es el que modela el problema a solucionar.

El diseño lógico se acerca más a la implementación del producto en una plataforma computacional tomando consideraciones de la plataforma en la cual va a ser implementado.

El diseño físico es la traducción del diseño conceptual en físico (realidad).

### El modelo de datos debe considerar

- Las entidades.
- Las relaciones entre las entidades.
- Las restricciones del modelo.
- Las suposiciones que se hicieron para modelar esta realidad.
- Se deben incorporar atributos.
- Se definen: dominios, identificadores, operaciones permitidas, entidades débiles y entidades fuertes.

**Algunos conceptos:***Atributo*

Algo que caracteriza a una entidad y está comprendido en su dominio. Cuando un atributo es identificador debe cumplir con las características de unicidad y minimalidad.

*Atributo compuesto*

Es aquel atributo que es suma de dos o más atributos de la entidad.

*Cardinalidad de un atributo con respecto a la entidad*

Indica el mínimo y el máximo de ocurrencias del atributo.

*Cardinalidad de la entidad respecto de la relación*

Indica el mínimo y el máximo de ocurrencias de una entidad respecto de otra entidad.

*Identificador*

Es algo que identifica de manera única a la entidad. Puede ser un atributo o una combinación de ellos. Si el atributo pertenece a la entidad es fuerte, si es externo es débil.

*Generalización*

Una entidad es una generalización de un grupo de entidades si cada una de éstas (subconjunto) es a su vez del tipo de entidad que generaliza.

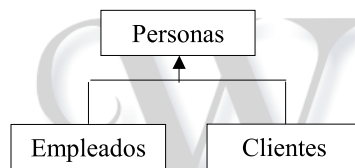
Se caracteriza por su cobertura, la cual puede ser total o parcial y exclusiva o superpuesta.

Total y exclusiva: todas las personas son empleados o clientes del banco, pero no ambas simultáneamente.

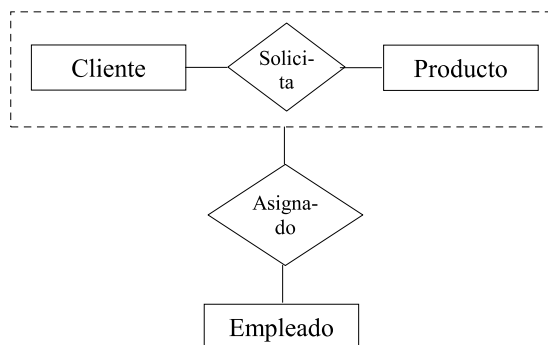
Total y superpuesta: todas las personas son empleados o clientes, permitiéndose que un empleado sea cliente.

Parcial y exclusiva: hay personas, algunas de las cuales son empleados o clientes.

Parcial y superpuesta: no todas las personas están clasificadas como empleados y clientes, y hay empleados que pueden ser clientes.

*Agregación*

Una relación y las entidades que relaciona puede ser manejada como una entidad, lo cual permite que se pueda relacionar con otras entidades.

**Estrategia para modelar con MER**

1. Identificar los tipos de entidad y las relaciones entre ellas.
2. Descomponer cada entidad para descubrir generalizaciones o agregaciones.
3. Verificar si no se subdividieron las relaciones.
4. Identificar los atributos, ya se de entidades o relaciones.
5. Definir los identificadores para las entidades.
6. Definir las restricciones de cardinalidad y cobertura.
7. Verificar el esquema resultante. Si no representa la realidad que quiero describir, se deben realizar modificaciones.

## INTRODUCCIÓN AL MODELO RELACIONAL

Persigue los siguientes objetivos:

1. **Independencia física:** es la capacidad de modificar el esquema interno sin tener que alterar el esquema conceptual (o los externos); es decir, el modo en que se almacenan los datos no influye en la manipulación lógica de los mismos. Por lo tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico. (Codd concede mucha importancia a este aspecto: *Independencia de ordenación*, *independencia de indexación* e *independencia en criterios de acceso*).
2. **Independencia lógica:** es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación. Se puede modificar el esquema conceptual para ampliar la base de datos o para reducirla, pero no repercute en los programas y/o usuarios que estén accediendo a sus conjuntos parciales de los mismos (vistas).
3. **Flexibilidad:** se le puede presentar al usuario los datos en la forma en que éste los prefiera o los requiera.
4. **Uniformidad:** las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
5. **Sencillez:** las características anteriores y los lenguajes sencillos de utilizar producen como resultado que el modelo de base de datos relacional sea fácil de comprender y utilizar por parte del usuario final.

Éste modelo se divide en tres partes: la estructura de los datos, la integridad de los datos y la manipulación de los datos.

### Estructura de los datos

Relación (tabla): Elemento básico del modelo relacional. Se asocia a una entidad. Todas las tablas en el modelo relacional son tablas planas, ésto es: el cruce de una fila y una columna solo puede dar un valor, es decir, no se admiten atributos multivaluados.

Tupla (fila o registro): Cada una de las instancias de una entidad es una tupla.

Atributo (camino o columna): Cada una de las propiedades que caracterizan a una entidad. Un atributo "A" es el rol que tiene un determinado dominio en la relación.

Cardinalidad (número de filas): Es el número de instancias de una entidad en una relación.

Grado (número de columnas): Número de atributos asociados a una entidad.

Clave primaria (identificador único): Conjunto de uno o más atributos que identifican de modo completo a una instancia de una entidad.

Dominio (conjunto de valores): Un dominio "D" es un conjunto finito de valores homogéneos y atómicos, caracterizados por un nombre. *Homogéneos:* los valores son todos del mismo tipo. *Atómicos:* son indivisibles, es decir, si se descomponen se pierde la semántica del dominio. Todo dominio tiene un nombre y un tipo de dato. A los dominios se les puede asignar medidas (kilos, metros, etc.) y otras restricciones. Los dominios no incluyen nulo, ya que nulo no es un valor.

La importancia del dominio es que restringe las comparaciones, es decir, solo se pueden comparar atributos definidos sobre el mismo dominio.

*Escalar:* Es la menor unidad semántica de información, es decir, es un dato atómico. Un dominio es un conjunto de escalares.

*Terminología:* Es aconsejable que el nombre de los dominios y de los atributos coincida siempre y cuando no exista ambigüedad.

*Dominio compuesto:* Combinación de dominios simples. Se les puede aplicar restricciones de integridad para evitar tener valores no válidos.

Relación: Matemáticamente una relación definida sobre los "n" dominios ( $D_1, D_2, \dots, D_n$ ), no necesariamente distintos, es un subconjunto del producto cartesiano de estos dominios, donde cada elemento de la relación (una tupla) es una serie de "n" valores ordenados. Esta definición (definición de Codd) no alude a atributos y dice que en las tuplas los valores están ordenados.

Podemos redefinir el concepto de relación diciendo: una relación R sobre un conjunto de dominios ( $D_1, D_2, \dots, D_n$ ), no necesariamente distintos, se compone de dos partes: una cabecera y un cuerpo. La cabecera está formada por un conjunto de atributos o, más precisamente, pares atributo-dominio. El cuerpo está formado por un conjunto de tuplas, las cuales están formadas por los pares atributo-valor.

$$\{ (A_1, D_1), (A_2, D_2), (A_n, D_n) \}$$

$$\{ (A_1, V_1), (A_2, V_2), (A_n, V_n) \}$$

La relación maneja dos conceptos: intención o esquema de relación y extensión o instancia de relación. La intención es la parte estática de la relación, la que se corresponde con la cabecera cuando la relación se percibe como una tabla. La extensión es el desarrollo de las tuplas de la relación. En el desarrollo de la extensión de la relación no hay dos tuplas iguales, el orden de las tuplas no es significativo, el orden de las columnas o atributos no es significativo y cada atributo solo puede tomar un único valor del dominio, no admitiendo, por lo tanto, grupos repetitivos. El orden de las tuplas no es significativo porque el acceso a la tabla es siempre a través de las claves definidas para esa relación; se asemeja a un registro secuencial. El orden de las columnas no es significativo porque todos los atributos de una tupla siempre están disponibles; no cambia el valor de la información esté en el orden que esté.

#### Interpretación de las tablas

1. Existe un dominio asociado a cada columna de la tabla.
2. Las dos partes de la tabla se corresponden con las dos partes de la relación.
3. Las filas se forman de pares atributo-valor.

#### Propiedades en general

1. No existen tuplas repetidas.
2. En dos o más tablas sí puede haber tuplas repetidas.
3. Las tuplas no están ordenadas.
4. Los atributos no están ordenados.
5. Los valores de los atributos son atómicos.

### **Integridad de los datos.**

#### Claves Primarias

*Superclave:* Es un conjunto de atributos que identifican de modo único a las tuplas de una relación. La mayor superclave es la suma de todos los atributos.

*Clave candidata:* Es un conjunto no vacío de atributos que identifican unívoca y mínimamente a cada tupla. Toda relación tendrá como mínimo una clave candidata. De todas las claves candidatas se va a elegir una que será la clave primaria de la relación.

*Unicidad:* no existen dos tuplas que posean el mismo valor en la clave candidata.

*Minimalidad:* no se puede eliminar ningún atributo de la clave candidata sin que se pierda la propiedad de unicidad.

*Clave primaria:* Aquella clave candidata que el usuario elegirá, por consideraciones ajenas al modelo relacional, para identificar las tuplas de una relación. Si la clave primaria es compuesta (más de un atributo), ésta debe ser no nula en su totalidad, es decir, ninguno de sus atributos puede ser nulo.

*Claves alternativas:* Aquellas claves candidatas que no han sido elegidas como clave primaria.

*Claves ajenas:* Una clave ajena es un conjunto de atributos de una relación R2 cuyos valores, o son completamente nulos o coinciden con los de la clave primaria de una relación R1. La relación donde se encuentra la clave ajena se denomina relación referencial y la relación donde se encuentra la clave primaria es la relación referida u objetivo. A su vez, una relación referida puede ser relación referencial en otra relación y viceversa.

Las claves ajenas pueden admitir nulos a diferencia de las claves primarias.

#### Reglas de integridad para las entidades

Ningún componente de la clave primaria de una relación puede aceptar nulos (ausencia de valor); es decir, en una base de datos relacional no se puede almacenar información sobre algo que no se puede identificar.

#### Reglas de integridad referencial

Toda clave ajena debe existir en el dominio de la clave primaria a la que está referenciando, es decir, no puede haber valores de claves ajenas sin concordancia con valores de claves primarias. Para cualquier valor no nulo de la clave ajena existe un valor asociado de la relación objetivo.

#### Reglas de integridad para las claves ajenas

*Borrado de claves primarias:*

- Restringido: no se puede borrar una tupla si está referenciada.
- Propagación o cascada: se borra la tupla junto con las tuplas que la están referenciando.
- Valor nulo: se borra la tupla y se pone valor nulo a todas las claves ajenas que la están referenciando.

*Modificación de claves primarias*

- Restringida: no se puede modificar una tupla si está referenciada.
- Propagación o cascada: se modifica la tupla junto con las tuplas que la están referenciando.
- Con valor nulo: se modifica la tupla y se pone valor nulo a todas las claves ajenas que la están referenciando.

**TEORÍA DE LA NORMALIZACIÓN**

Con el MER se pueden presentar los siguientes problemas:

- Incapacidad para almacenar ciertos hechos.
- Redundancias → posibilidad de incoherencia.
- Ambigüedades.
- Pérdida de información.
- Pérdida de dependencias funcionales.
- Aparición de estados no válidos en la base de datos, es decir, anomalías de inserción, borrado y modificación.

En conclusión, el MER obtenido siempre debe ser analizado para comprobar que no presente los problemas anteriores. Los problemas mencionados suceden básicamente por no cumplir con el enunciado que debe regir en todo diseño: hechos distintos deben almacenarse en objetos distintos. Una forma de evitar estos problemas es seguir un riguroso diseño conceptual y un traspaso de éste al modelo relacional.

La normalización penaliza las consultas al disminuir la eficiencia ya que la normalización aumenta el número de relaciones o tablas presentes en la base de datos. Por lo que una determinada consulta puede implicar el acceso a varias tablas, aumentando el costo de éstas.

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.
- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

**Primera forma normal (por Codd).**

Es una restricción inherente al modelo relacional por lo que su cumplimiento es obligatorio. Consiste en la prohibición de que en una relación existan grupos repetitivos, es decir, un atributo no puede tomar más de un valor de su dominio.

Código	Nombre	Cursos
1	Marcos	Inglés
2	Lucas	Contabilidad, Informática
3	Marta	Inglés, Contabilidad

*No está normalizada*

Se soluciona dividiendo la tabla en:

Código	Nombre
1	Marcos
2	Lucas
3	Marta

Código	Curso
1	Inglés
2	Contabilidad
2	Informática
3	Inglés
3	Informática

### Segunda forma normal (por Codd).

Una tabla está en segunda forma normal si, además de estar en primera forma normal, todos los atributos que no forman parte de ninguna clave candidata suministran información acerca de la clave completa.

Código Empleado	Código Dpto.	Nombre	Departamento	Años
1	6	Juan	Contabilidad	6
2	3	Pedro	Sistemas	3
3	2	Sonia	I+D	1
4	3	Verónica	Sistemas	10
2	6	Pedro	Contabilidad	5

*No está normalizada*

Solución

Tabla A		Tabla B	
Código Empleado	Nombre	Código Departamento	Dpto.
1	Juan	2	I+D
2	Pedro	3	Sistemas
3	Sonia	6	Contabilidad
4	Verónica		

Tabla C		
Código Empleado	Código Departamento	Años
1	6	6
2	3	3
3	2	1
4	3	10
2	6	5

### Tercera forma normal.

Se dice que una tabla está en tercera forma normal si y solo si los campos de la tabla dependen únicamente de la clave primaria.

Código	Nombre	Curso	Aula
1	Marcos	Informática	Aula A
2	Lucas	Inglés	Aula B
3	Marta	Contabilidad	Aula C

*No está normalizada*

Solución:

Tabla A			Tabla B	
Código	Nombre	Curso	Curso	Aula
1	Marcos	Informática	Informática	Aula A
2	Lucas	Inglés	Inglés	Aula B
3	Marta	Contabilidad	Contabilidad	Aula C

#### Cuarta forma normal.

Una tabla está en cuarta forma normal si y sólo si para cualquier combinación clave - campo no existen valores duplicados. Veámoslo con un ejemplo:

*Geometría*

Figura	Color	Tamaño
Cuadrado	Rojo	Grande
Cuadrado	Azul	Grande
Cuadrado	Azul	Mediano
Círculo	Blanco	Mediano
Círculo	Azul	Pequeño
Círculo	Azul	Mediano

*No está normalizada*

Solución

Figura	Tamaño	Figura	Color
Cuadrado	Grande	Cuadrado	Rojo
Cuadrado	Pequeño	Cuadrado	Azul
Círculo	Mediano	Círculo	Blanco
Círculo	Pequeño	Círculo	Azul