

Base de datos

Dos tipos fundamentales de sistemas de información electrónico.

Sistemas de gestión de ficheros: Forma que inicialmente se desarrolló en la informática para gestionar ficheros. Los datos se almacenan en ficheros y se crean aplicaciones para acceder a los ficheros. Cada aplicación organiza los datos en los ficheros como le parece mejor. **Los procesos son independientes por lo que la modificación de uno no afecta al resto.**

Programación de aplicaciones compleja. los programadores se encargan de lo que tiene que hacer la aplicación

Datos redundantes. Se repiten continuamente.

Datos inconsistentes. Un proceso cambia sus datos y no los del resto. Por lo que la misma información puede tener distintos valores según qué aplicación acceda a él.

Difícil acceso a los datos. Una consulta no prevista, hay que modificar el código de las aplicaciones o crear una'.

Coste de almacenamiento elevado. Al guardar varias veces el mismo dato, se requiere más espacio en los discos.

Dificultad para el acceso simultáneo a los datos. El acceso simultáneo requiere que varios usuarios al puedan acceder a la misma información, es difícil conseguir esta capacidad.

Sistemas de bases de datos: Los datos se centralizan en una base de datos común a todas las aplicaciones. El software **Sistema Gestor de Bases de Datos (SGBD)** es el que accede a los datos y se los gestiona. Las aplicaciones que creen los programadores, no acceden directamente a los datos.

la base de datos en sí solo consta de los propios datos en forma estructurada, mientras que el SGBD es el elemento básico para materializar estas estructuras.

Independencia de los datos y los programas. modifica los datos sin modificar el código de las aplicaciones

Menor redundancia. No requiere que los datos se repitan para cada aplicación que los requiera. Los programadores deberán conocer la estructura creada para los datos y la forma en la que deben acceder a ellos.

Integridad de los datos. Al estar centralizados, es más difícil que haya datos incoherentes.

Mayor seguridad en los datos. El SGBD es el encargado de la seguridad.

Visiones distintas según el usuario. Nuevamente, centralizar los datos facilita crear políticas que permitan que los usuarios vean la información de la base de datos de forma distinta.

Datos más documentados. Las bases de datos tienen mucho mejor gestionados los metadatos, que permiten describir la información de la base de datos y que pueden ser consultados por las aplicaciones

Acceso a los datos más eficiente. Esta forma de organizar los datos produce un resultado más óptimo en rendimiento ya que los sistemas gestores centralizan el acceso pudiendo ejecutar políticas diferentes en función de la demanda.

Menor espacio de almacenamiento. Puesto que hay muy poca redundancia.

Acceso simultáneo a los datos.. Cuando varias aplicaciones intentan acceder a los datos, compiten por los datos y es fácil el bloqueo mutuo. Toda petición pasa la capa del SGBD y esto permite evitar los bloqueos.

Instalación costosa. El control y administración de bases de datos requiere de un software y hardware poderoso.

Requiere personal cualificado. Debido a la dificultad de manejo de este tipo de sistemas.

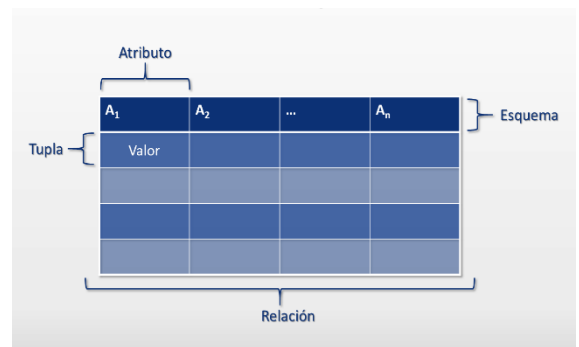
Implantación larga y difícil. La adaptación del personal y del equipamiento es mucho más complicada y lleva bastante tiempo.

Sistemas de base de datos relacional: Basado en el concepto de tablas y las relaciones entre ellas como forma de relacionar la información entre sí. Los datos se organizan en tablas y estas en columnas y filas de datos. El lenguaje habitual es SQL un estándar.

Se llamará **registro, entidad o tupla** a cada fila de la tabla y **campo o atributo** a cada columna de la tabla.

Una **clave** será un atributo o conjunto de atributos que identifique de forma única a una tupla.

Funcionalidades:



Tipos de datos: Cada columna tiene un tipo de dato definido de manera que el SGBD no permite almacenar valores de otro tipo en dicha columna.

Restricciones: Es posible definir restricciones que obliguen a cumplir una serie de requisitos a los valores que se almacenan en una columna determinada.

Integridad referencial: En el momento de registrar algún nuevo dato que deba estar relacionado con otro, el SGBD comprobará que el segundo existe antes de permitir el registro.

Consultas complejas: Es posible realizar consultas muy complejas.

Ventajas:

Menor redundancia. No hace falta tanta repetición de datos.

Menor espacio de almacenamiento. Gracias a una mejor estructuración de los datos.

Acceso a los datos más eficiente. La organización de los datos produce un resultado más óptimo en rendimiento.

Integridad de los datos. Mayor dificultad de perder los datos o de realizar incoherencias con ellos.

Mayor seguridad en los datos. Al limitar el acceso a ciertos usuarios.

Desventajas:

Instalación costosa. El control y administración de bases de datos requiere de un software y hardware potente.

Requiere personal cualificado. Debido a la dificultad de manejo de este tipo de sistemas.

Implantación larga y difícil. La adaptación del personal es mucho más complicada y lleva bastante tiempo.

Las Bases de Datos relacionales son apropiadas si:

Es necesario realizar consultas muy complejas entre diferentes tablas

Es necesario validar la información entre tablas

Se permite que un dato pueda tener cualquier número de valores

Es necesario construir nuevas consultas que no habían sido planificadas cuando se diseñó la Base de Datos

Los SGBD tienen que realizar tres tipos de funciones para ser considerados válidos, para ello utilizaremos los siguientes 3 lenguajes:

Descripción o definición mediante DDL, manipulación mediante DML, control mediante DCL.

función de definición (DDL)

Crear todos los elementos estructurales de la base de datos

Especificar reglas especiales que deben cumplir los datos

Relacionar los datos de forma precisa

Especificar el significado de los datos

Función de manipulación (DML)

Cambia y consulta los datos de la base de datos.

Añadir datos

Eliminar datos

Modificar datos

Consultar datos

Función de control (DCL)

Permite al administrador gestionar el acceso a los datos.

Se encarga de los permisos de acceso a los elementos que forman parte de la base de datos.

Niveles de abstracción

En cualquier sistema de información se considera que se pueden observar los datos desde dos puntos de vista:

Vista externa. Esta es la visión de los datos que poseen los usuarios del Sistema de Información.

Vista física. Visión de los creadores del software, que determina su forma de funcionar.

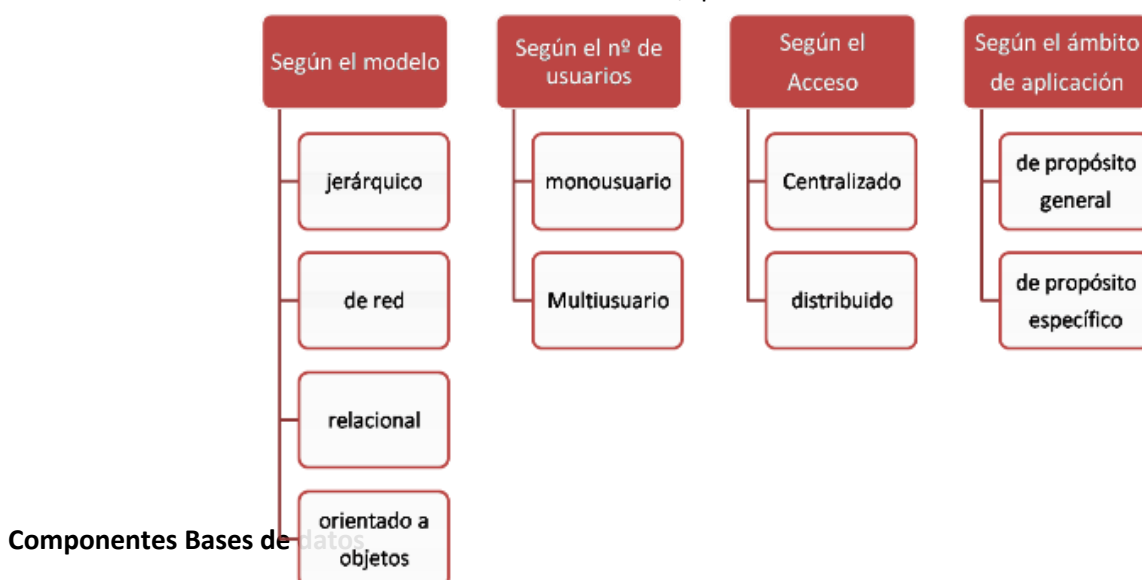


Tabla: Conjunto de filas y columnas bajo un mismo nombre, con valores almacenados para una serie de datos.

Campo: Las columnas de una tabla. Identifica una familia de datos.

Informe: Es un listado ordenado de los campos y registros seleccionados en un formato fácil de leer.

Guiones o scripts: Son un conjunto de instrucciones, que ejecutadas, realizan operaciones avanzadas o mantenimiento en la BD.

Procedimientos: Son un tipo especial de script que están almacenados en la BD y forman parte de su esquema.

Registro: Corresponde a cada una de las filas de la tabla. También se llaman tuplas.

Índice: Es una estructura que almacena los campos clave de una tabla

Consulta: Es una instrucción para hacer peticiones a una BD.

Tipo de Dato: Indica la naturaleza del campo, que son los que contienen caracteres alfabéticos y números...

Vista: Se obtienen al guardar una consulta de una o varias tablas.

Una base de datos almacena los datos a través de un esquema. El esquema es la definición de la estructura donde se almacenan los datos

El modelado conceptual o diseño conceptual, constituye la primera fase de desarrollo de bases de datos, y puede subdividirse en dos etapas: – Análisis de Requisitos – Generación del esquema conceptual (Conceptualización).

-Definen cómo se modela la estructura lógica de una base de datos

-Son entidades fundamentales para introducir la abstracción en una base de datos

-Definen como los datos se conectan entre sí y como se procesan y almacenan dentro del sistema.

-Su enfoque principal es apoyar y ayudar a los sistemas de información mostrando el formato y la definición de los diferentes datos involucrados.

-También ayudan a evitar la redundancia de datos

Los tres tipos básicos de modelo de datos son: de datos conceptuales, de datos físicos y de datos lógicos

Un modelo conceptual de datos identifica las relaciones de más alto nivel entre las diferentes entidades.

Las características del modelo conceptual de datos incluyen:

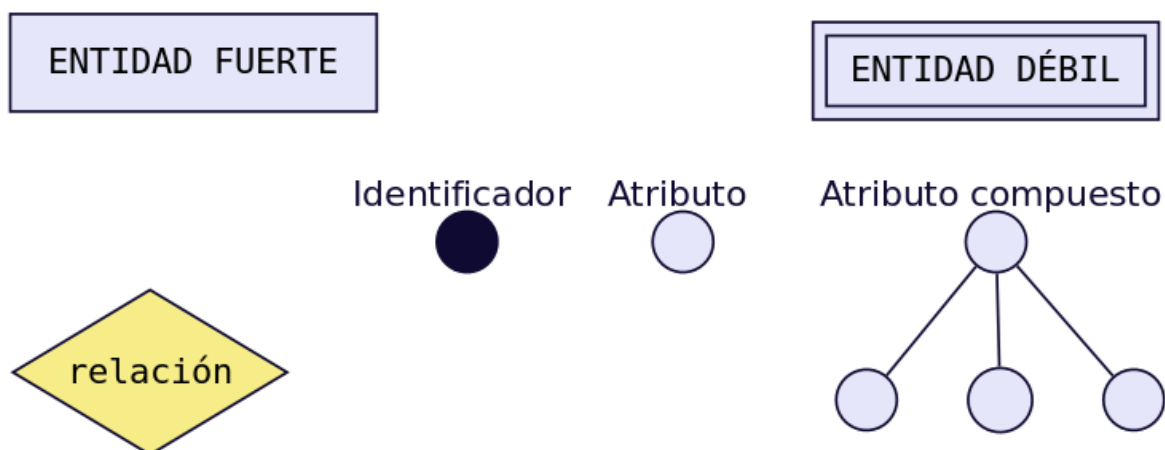
Incluye las entidades importantes y las relaciones entre ellas, no se especifica ningún atributo y no se especifica ninguna clave principal.

Los pasos que están asociados al diseño conceptual son:

- Identificar los tipos de **entidad**.
- Identificar los tipos de **relación**.
- Identificar y asociar los **atributos** con los tipos de entidad y de relación.
- Determinar los **dominios** de los atributos.
- Determinar los **atributos de clave candidata, principal y alternativa**.
- Comprobar si el modelo tiene **redundancia**.
- Dibujar el **diagrama** entidad-relación

Repasar el modelo de datos conceptual con los usuarios

El modelo Entidad-Relación es el modelo más utilizado para el diseño conceptual de bases de datos. Se basa en la representación de objetos a los que se les da el nombre de entidades, y asociaciones entre ellos, llamadas relaciones.

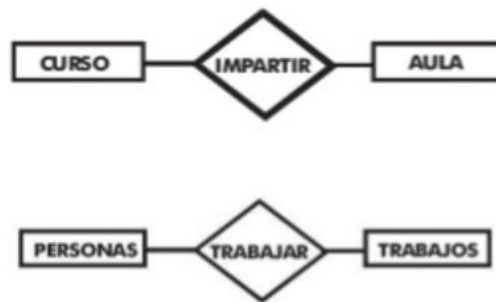


Una entidad es cualquier objeto o elemento acerca del cual se pueda almacenar información en la BD. Las entidades pueden ser concretas como una persona o abstractas como una fecha.

Hay dos tipos de entidades: fuertes y débiles. Una entidad débil es una entidad cuya existencia depende de la existencia de otra entidad. Una entidad fuerte es una entidad que no es débil.

Los atributos definen o identifican las características de entidad (**es el contenido de esta entidad**). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Siempre debe existir un atributo identificativo.

- Correspondencia o asociación entre dos o más entidades.
- Cada relación tiene un nombre que describe su función.



Las **relaciones**. Las entidades que están involucradas en una determinada relación se denominan **entidades participantes**. El número de participantes en una relación es lo que se denomina **grado de la relación**.

- **Relaciones Reflexivas:** consiste en relacionar una entidad consigo misma.

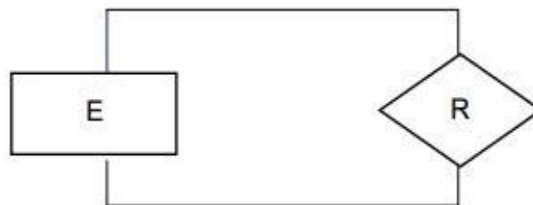


Figura 1. Relación Reflexiva

- **Relaciones binarias:** consiste en unir dos entidades mediante una relación.

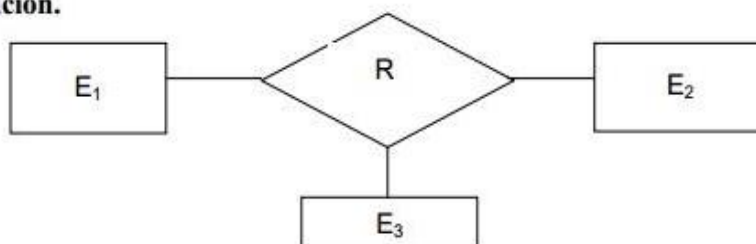
• **Cal**



Figura 2. Relación binaria.

ntidad




- **Relaciones ternarias:** Son aquellas que unen tres entidades mediante una relación.



de sus
entidad

Figura 3. Relación Ternaria


- Caso contrario a la Total.


TIPO	RELACIÓN	REPRESENTACIÓN
1:1	Uno a uno: La cardinalidad máxima en ambas direcciones es 1.	1  1
1:N	Uno a muchos: La cardinalidad máxima en una dirección es 1 y en la otra muchos.	1  N
N:M	Muchos a muchos: La cardinalidad máxima en ambas direcciones en muchos.	N  M

generalización \equiv abstracción que conduce a la definición un nuevo tipo de entidad (supertipo) a partir de las “similitudes” de otros (subtipos)

especialización \equiv refinamiento que permite obtener nuevos tipos de entidad (subtipos) a partir de las “diferencias” entre las ocurrencias de un tipo (supertipo)

- ✓ la cardinalidad es siempre $(1, 1)$ en el supertipo y $(0, 1)$ ó $(1, 1)$ en los subtipos
- ✓ los subtipos heredan los atributos del supertipo \rightarrow jerarquía de entidades
- ✓ se denota con triángulo con base paralela al supertipo (+ atributo selector, si existe)
- ✓ se pueden añadir propiedades de (exclusión/solapamiento) y (totalidad/parcialidad)

 arco

 círculo en arco

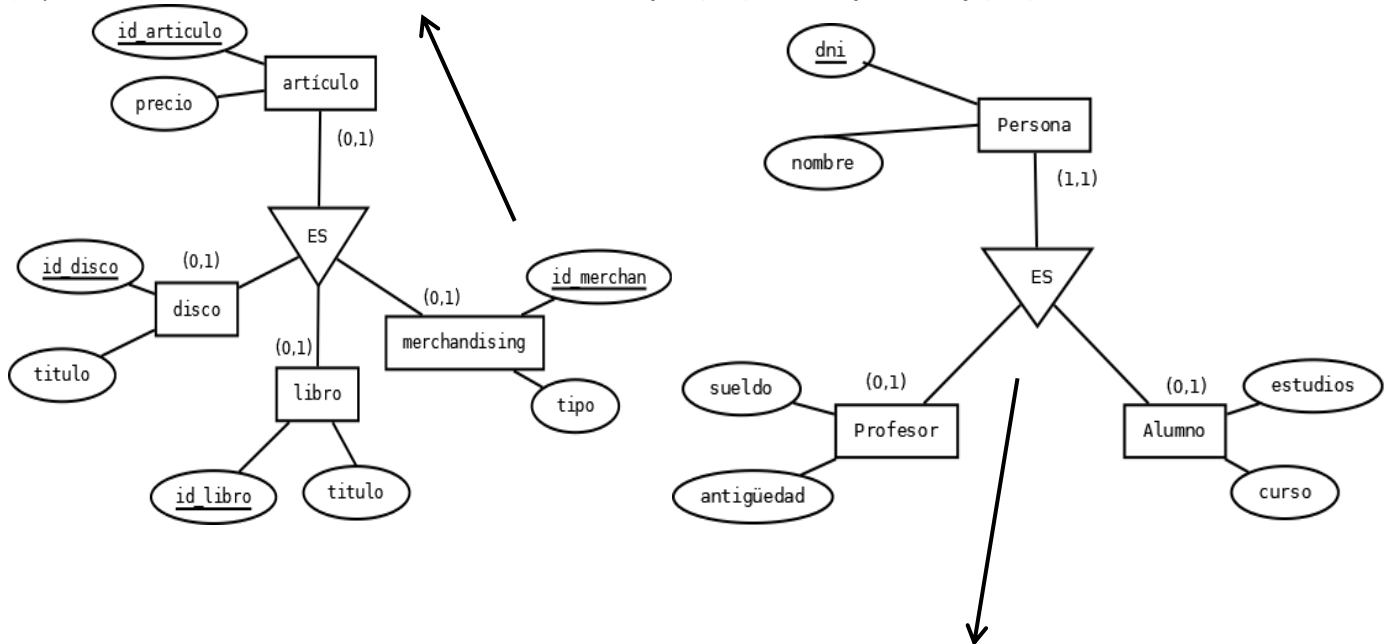
↑ Generalización

- Énfasis en las **similitudes**
- Cada **instancia** del **supertipo** es **también** una **instancia** de alguno de los **subtipos**

↓ Especialización

- Énfasis en las **diferencias**
- Alguna **instancia** del **supertipo** puede **no ser** instancia de ningún subtipo

Generalización: Es una relación contenida que existe entre el conjunto de nivel más alto (**superclase**) y uno o más conjuntos de entidades de nivel más bajo (**subclase**). Permite que las entidades de nivel más bajo **hereden los atributos** de la entidad generalizadora de más alto nivel. La Generalización trata de **eliminar la redundancia** (repetición) de atributos. **Las cardinalidades son siempre (1,1) en la superclase y (0,1) en la subclase.**

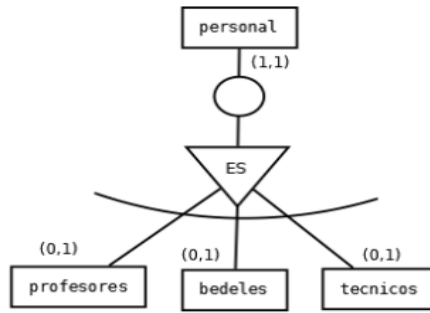
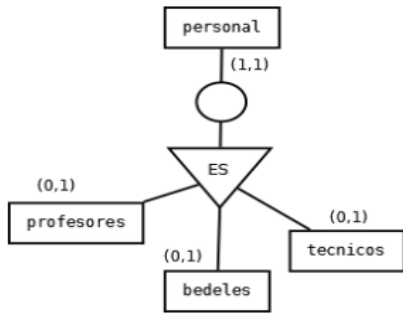


Especialización: Es el proceso por el que se definen las diferentes subclases de una superclase se conoce como especialización. Es la operación inversa a la generalización.

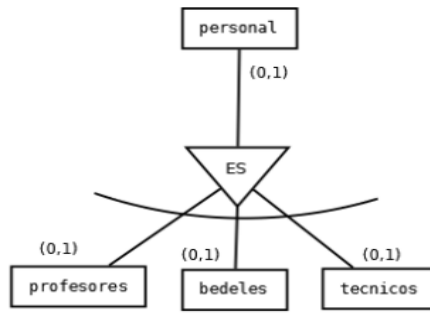
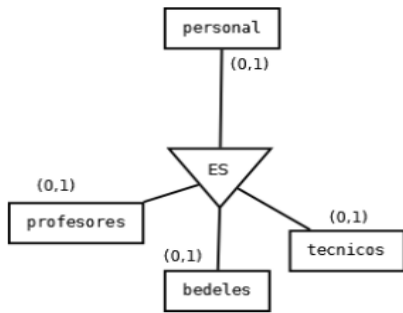
La jerarquía es el proceso de subdividir una entidad en varias subentidades relacionándolas con la entidad a la que se refieren. Puede haber dos tipos: **Total** **Parcial**

Y los dos tipos de subentidades que puede haber, se dividen en dos también: **Exclusiva** **Inclusiva**

Según el máximo número de ocurrencias de una subentidad que se corresponden con una ocurrencia de la superentidad.



Total Solapada y Total Exclusiva respectivamente.



Parcial Solapada y Parcial Exclusiva respectivamente.

Exclusivo es uno de los tipos.

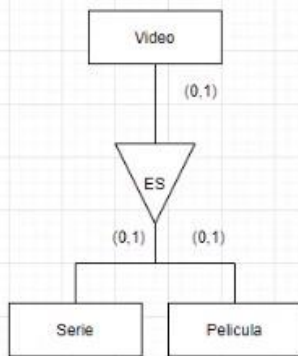
Parcial una más de las que están

Inclusivo más de uno de los tipos

Total una de las que están ahí.

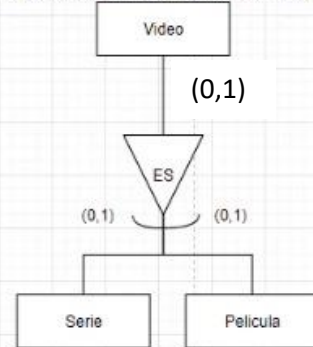
Inclusiva parcial

1 Video, puede ser además de ser serie o película, puede ser otra cosa. (documental)



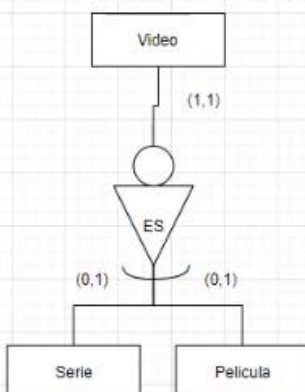
Exclusiva y parcial

1 video SOLO puede ser o serie o película. U otra cosa (documental). Solo puede ser 1 de las tres, por ejemplo



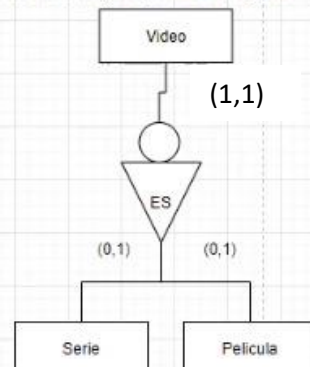
Exclusiva y Total

1 video, tiene que ser o una serie o una película y NO puede ser las dos cosas a la vez



Inclusiva y Total

1 video puede ser a la vez una serie y una película. Y obligatoriamente SERIE O PELICULA.

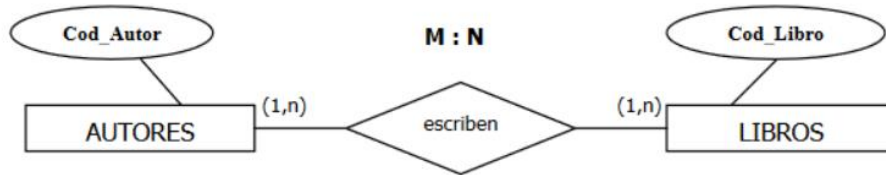


Transformadas

La transformación se realiza empleando las siguientes reglas:

Toda entidad se transforma en una tabla. Todo atributo se transforma en columna dentro de la tabla. El identificador único de la entidad se convierte en clave primaria.

Ejemplo: Relación entre los autores y los libros que escriben.



escriben (Cod_Autor, Cod_Libro,)

AUTORES (Cod_Autor,)

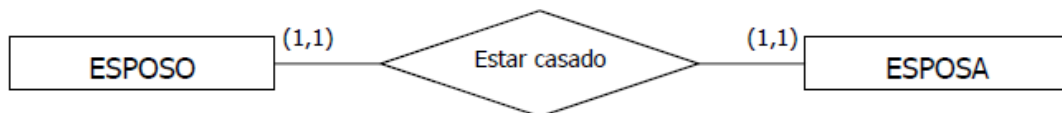
LIBROS (Cod_Libro,)



LIBRO (Cod_Libro, Título, ... , Cod_Editorial)

EDITORIAL (Cod_Editorial, Nombre,)

1 : 1



Solución a)

ESTAR CASADO (Cod_esposo,, Cod_esposa,)

Solución B)

ESPOSO (Cod_esposo,, Cod_esposa)

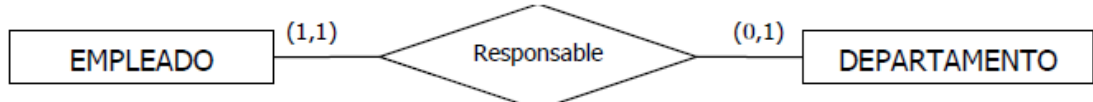
ESPOSA (Cod_esposa,)

Solución C)

ESPOSO (Cod_esposo,, Cod_esposa)

ESPOSA (Cod_esposa,, Cod_esposo)

1 : 1



EMPLEADO (Cod_empleado)

DEPARTAMENTO (Cod_departamento,, Cod_empleado)

↓
Clave ajena not null

Modelo Entidad Relación	Modelo Relacional
Entidad	Tabla
Atributo	Columna/Campo
Identificador único	Clave Primaria
Relación 1:N	Se toma la llave primaria de la tabla 1 y se inserta en la tabla de muchos (siendo la llave foránea)
Relación N:M	Se crea una nueva tabla y la llave primaria de ambas tablas estarán en la tabla creada como llaves foráneas.
Relación 1:1	Se toma la llave primaria de la tabla fuerte y se ingresa en la tabla débil (siendo llave foránea)
Generalización/Especialización	La llave primaria de la tabla dominante pasara como llave foránea en las tablas dependientes.

