

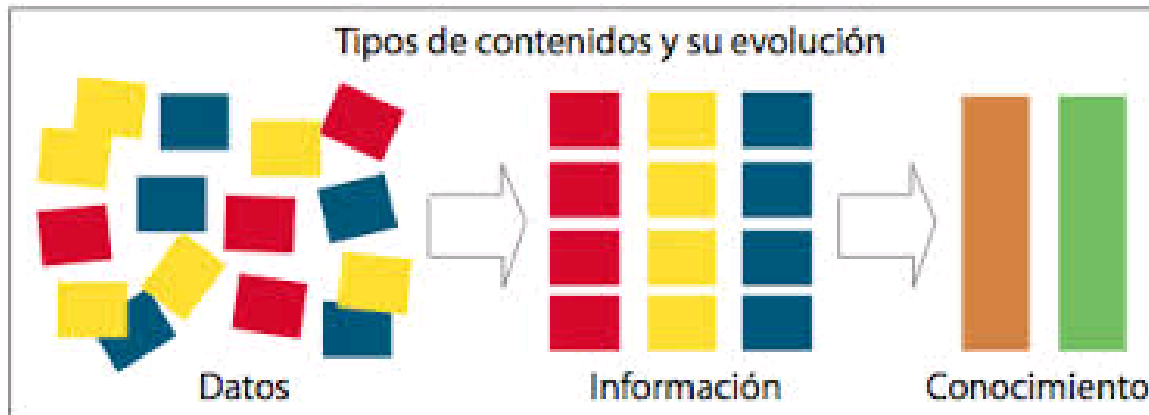
CODES

Capacitación Base de Datos - Conceptos

Dato: es una representación simbólica (numérica, alfabética, algorítmica, etc...), un atributo o característica de una entidad. Los datos son hechos que describen sucesos y entidades. No tienen ninguna información.

Información: es un conjunto de datos significativos y pertinentes que describen sucesos o entidades. Para ser significativos, los datos deben constar de símbolos reconocibles, estar completos y expresar una idea no ambigua.

Podemos decir que Información son Datos que han sido procesados de tal forma que son significativos para quien los utiliza.



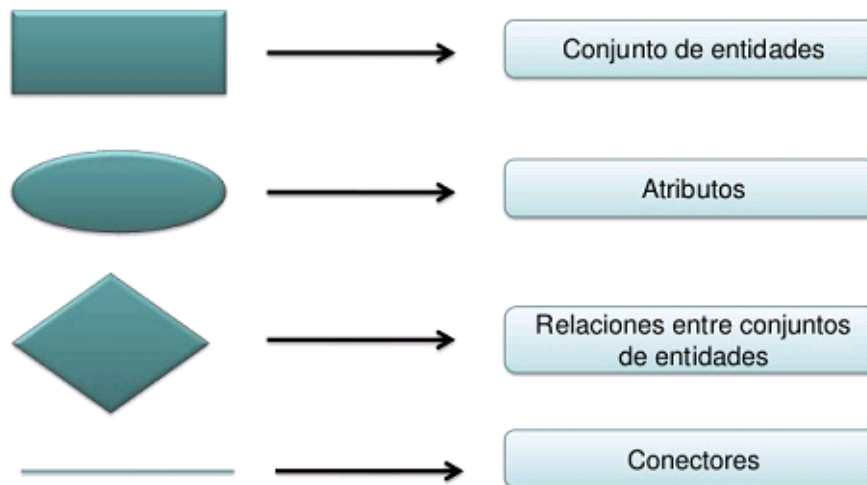
ENTIDAD

Definición general: “Todo aquello del mundo que nos interesa”. Se debe definir cual es el ámbito de interés.

Los objetos que aparecen en la vida real, son los que llamamos entidades. Por ejemplo, alumnos, empleados, aviones, coches, alojamientos,...

- Una entidad da lugar a una o n tablas en una Base de Datos.

MODELO ENTIDAD - RELACION

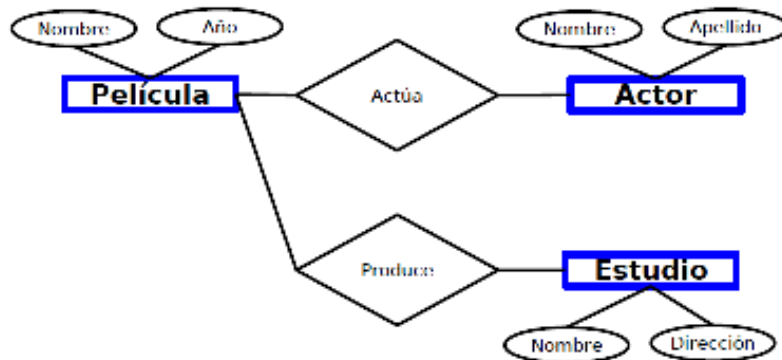


ATRIBUTOS:

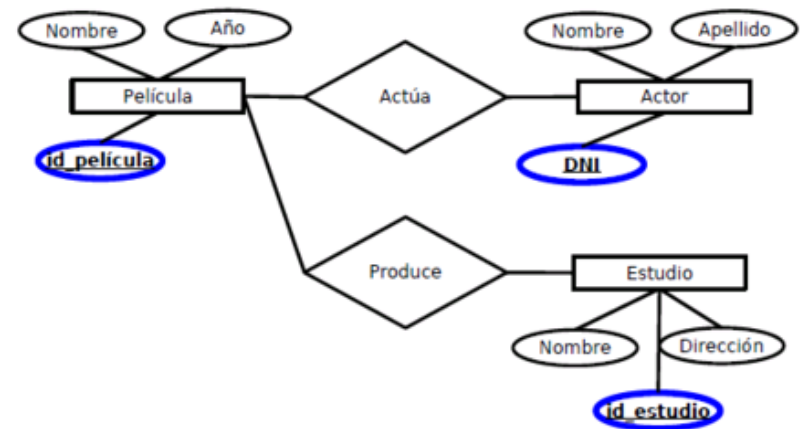
Las entidades están compuestas por varios atributos, que vienen a ser sus propiedades. Por ejemplo, la entidad alumnos, tendrá los atributos nombre, DNI, nacionalidad, fecha de nacimiento, ...

Los atributos califican a una entidad.

Entidades



Atributos clave: subrayados



Elementos u Ocurrencias

Cada entidad tendrá un número ilimitado de elementos. Por ejemplo, un elemento de la entidad alumnos será un alumno en sí; así el alumno Pepe será un elemento, José será otro, ...

Cada uno de esos elementos también recibe el nombre de fila o registro en la terminología de Base de Datos.

Combinando estos tres conceptos tenemos una estructura del tipo tabla en la terminología de Base de Datos.

Tabla ALUMNOS:

Nombre	DNI	Nacionalidad	Fecha nacim.
Joseph	99.999.999	Irlanda	2-2-1969
Pepe	88.888.888	España	8-8-1968
José	77.777.777	Chile	7-7-1967
...			

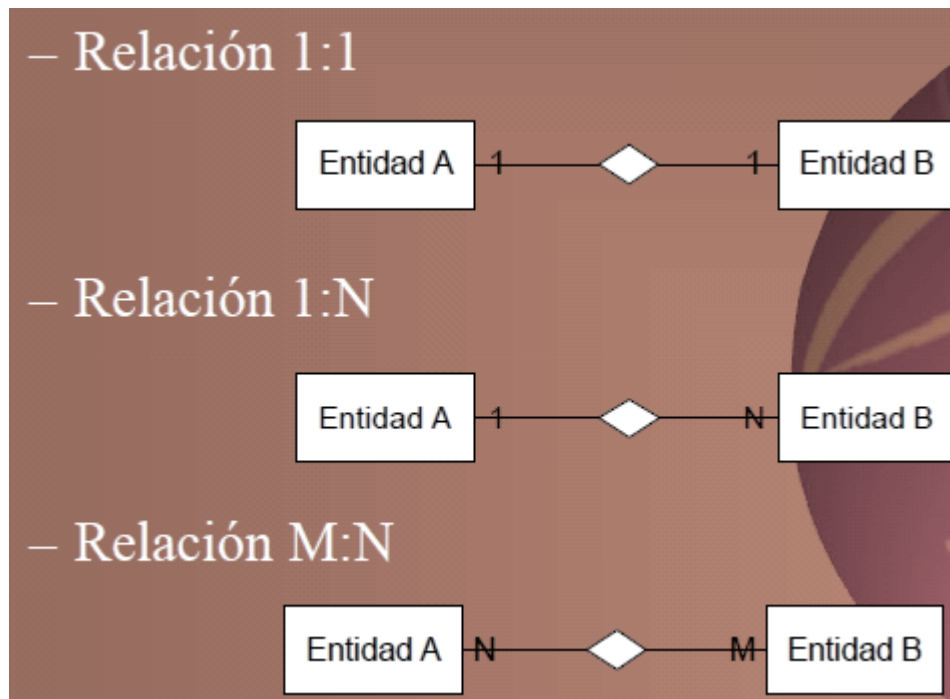
Diagram illustrating the structure of the ALUMNOS table. The table has four columns: Nombre, DNI, Nacionalidad, and Fecha nacim. The rows represent individual students, with examples: Joseph (Irlanda, 2-2-1969), Pepe (España, 8-8-1968), and José (Chile, 7-7-1967). The label 'Filas' (Rows) points to the rows of data, and the label 'Columnas' (Columns) points to the columns of data.

Relaciones

Las entidades no están aisladas sino que están relacionadas entre sí.

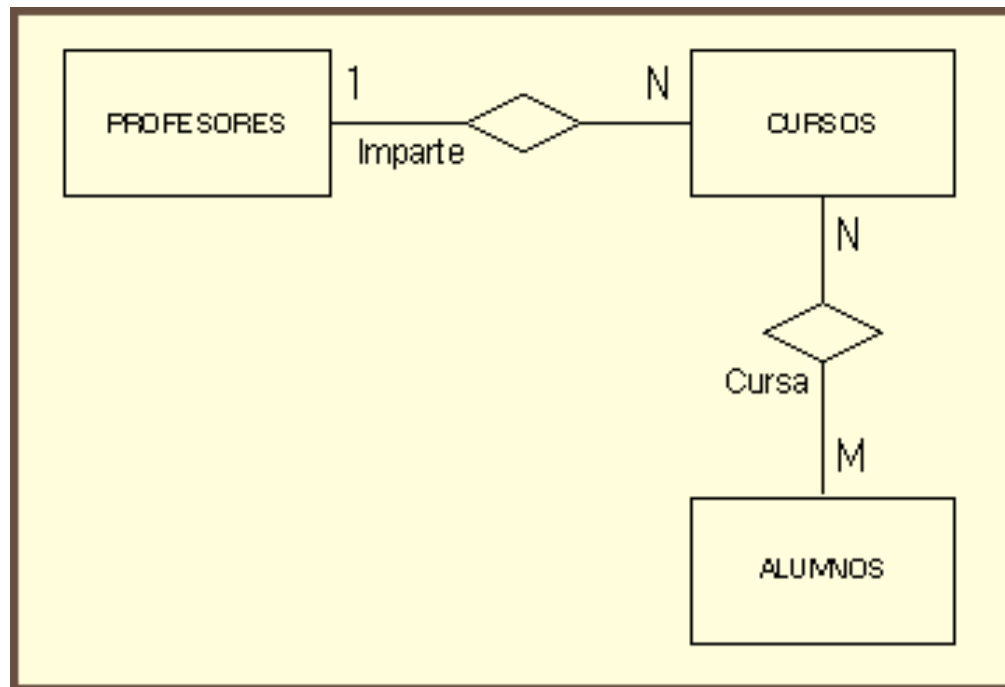
Estas relaciones pueden ser de tres tipos diferentes:

- 1 a 1
- 1 a muchos (1 a N)
- Muchos a muchos (M a N)

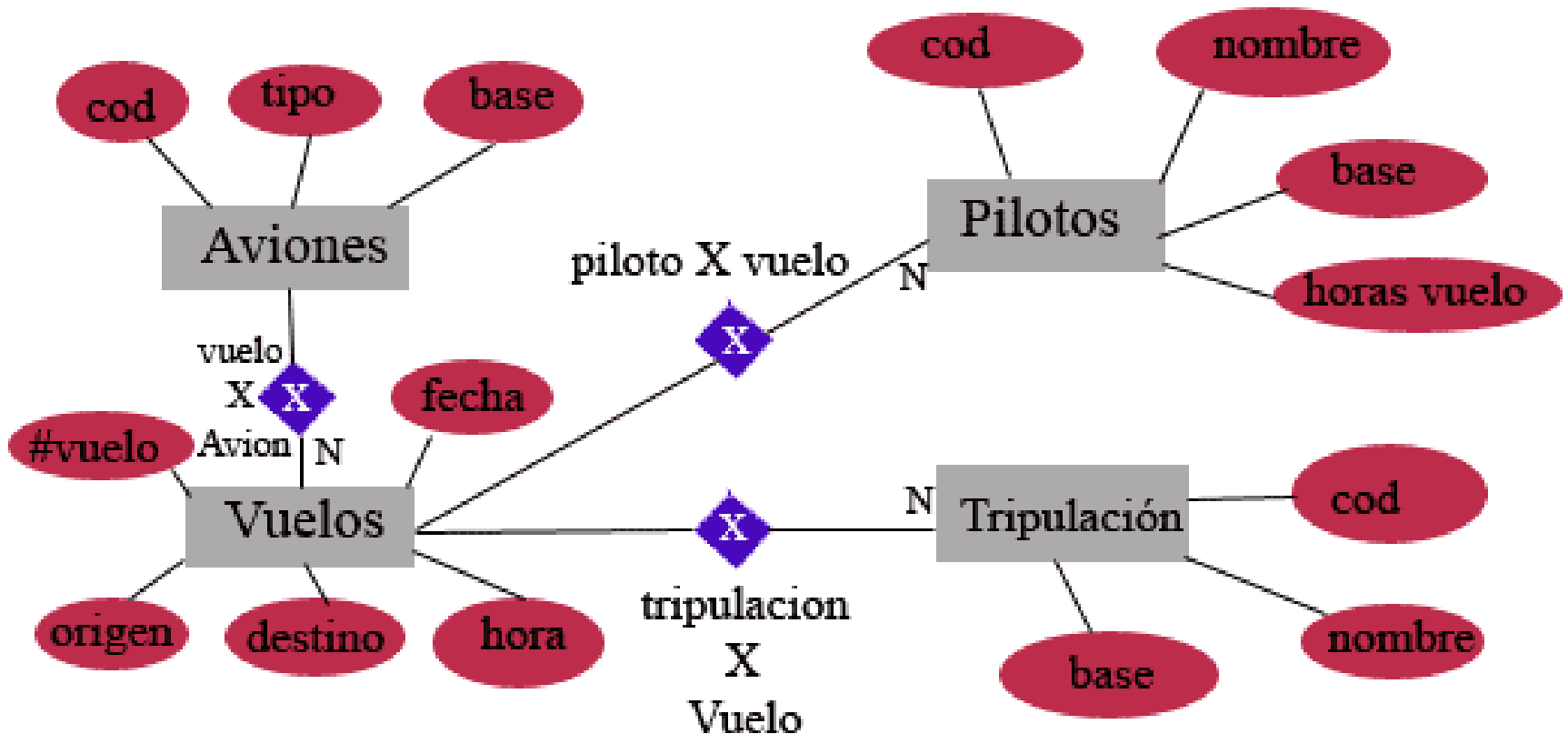


Ejemplo de Relación

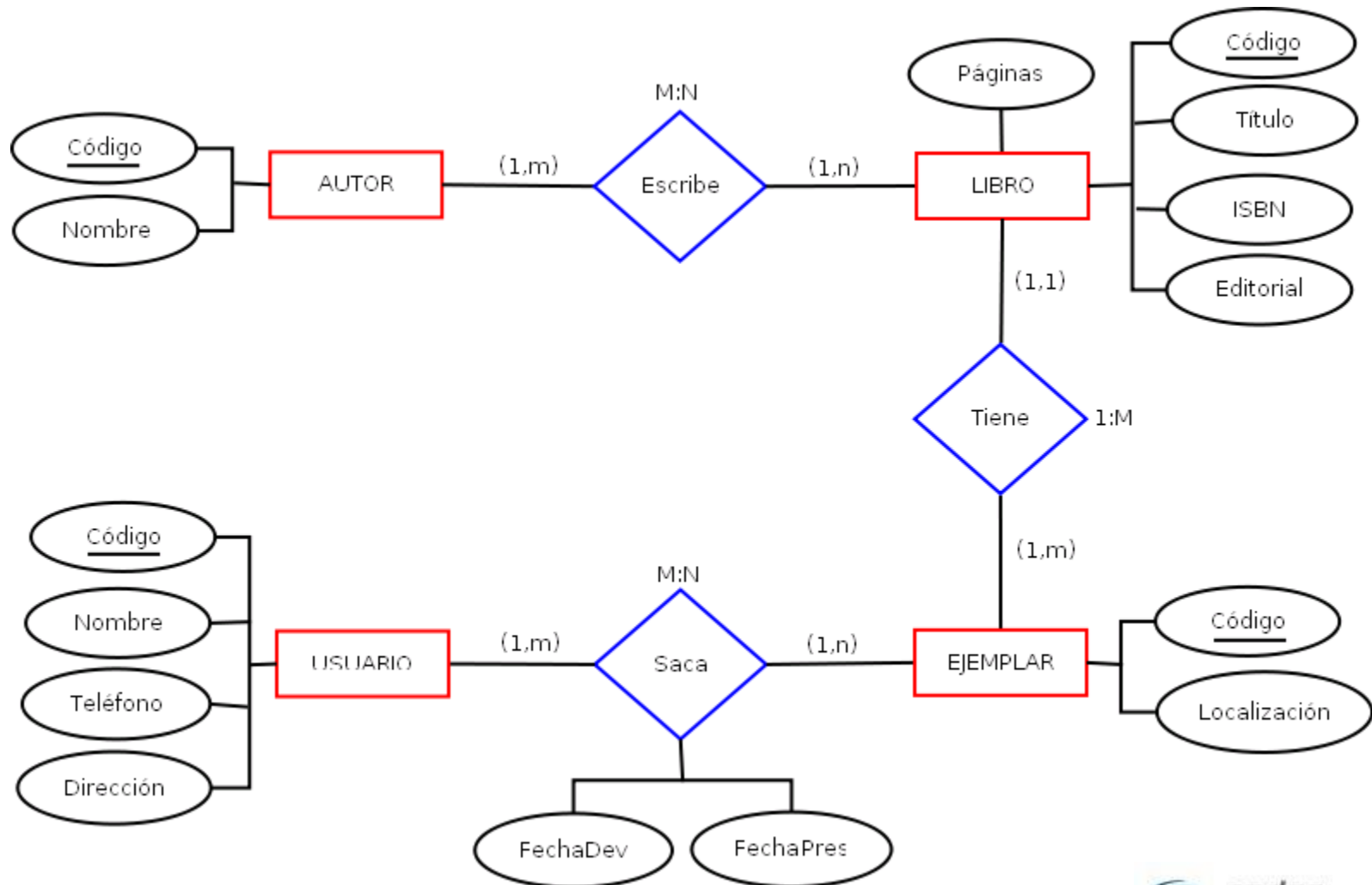
Hacer un modelo Entidad-Relación para la siguiente situación:
Tenemos una universidad, en la que hay varios cursos. Cada curso está dirigido por un profesor, el cual puede dirigir varios cursos. Los alumnos pueden inscribirse en uno o más cursos.



Entidades, Relaciones y Atributos – Ejemplo Vuelos Comerciales



Entidades, Relaciones y Atributos – Ejemplo Biblioteca.



Definición de Base de Datos

Son datos organizados y relacionados entre si, los cuales son recolectados y explotados por los sistemas de una empresa o negocio en particular.

Para poder acceder a una base de datos es necesario tener un Sistema Gestor de Base de Datos (SGBD).

Otra definición: Una Base de Datos (BD) es un conjunto de datos interrelacionados almacenados en conjunto, sin redundancias innecesarias, de forma independiente de los programas que acceden a ellos..



Sistema Gestor de Base de Datos.

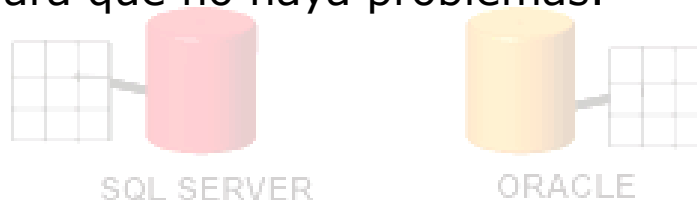
(Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.

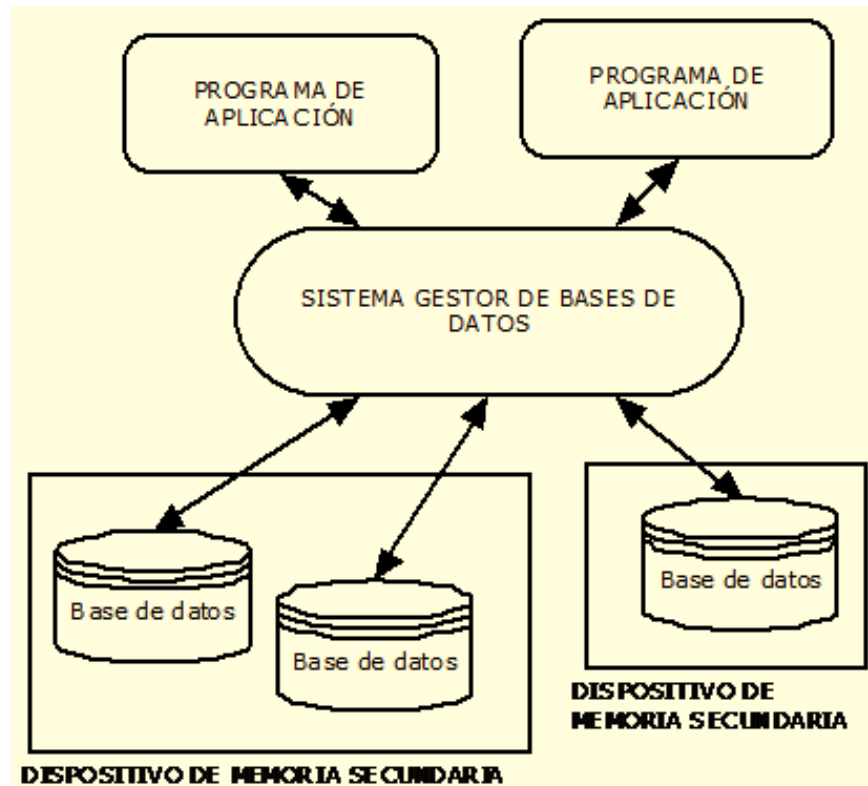
Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de almacenamiento.

Manipular una base de datos: incluye funciones como consulta, actualización, etc. de bases de datos. Si el sistema soporta bases de datos relacionales se llama RDBMS en inglés o SGBDR en español. Otras funciones de los SGBD:

- ✓ En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos. Controlar la concurrencia implica que si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas.



- ✓ Un SGBD también debe encargarse de cumplir las reglas de integridad y redundancias.
- ✓ Otra función importante en un SGBD es su capacidad de realizar copias de seguridad y de recuperación de datos.
- ✓ Restricción de accesos no autorizados.
- ✓ Suministrar múltiples interfaces de usuario.

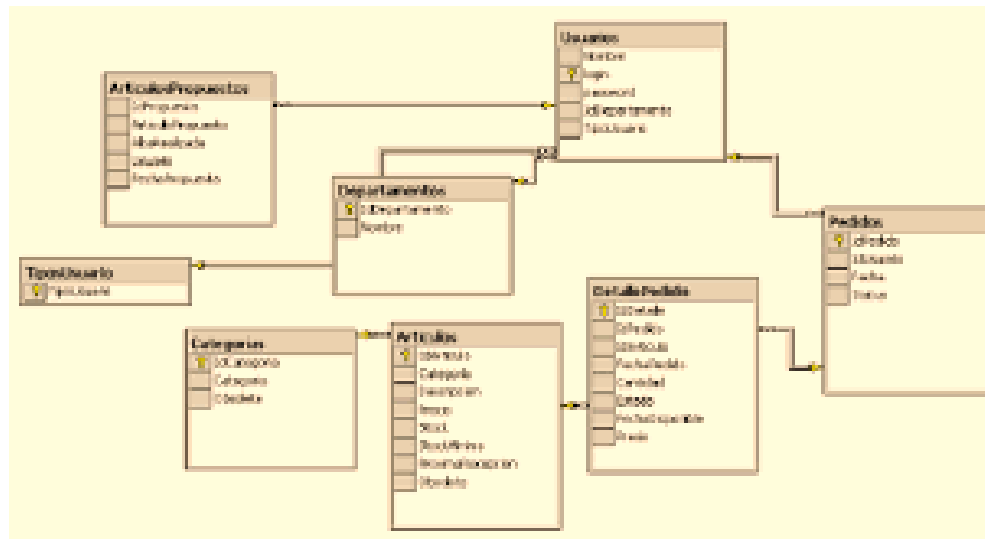


Las Bases de Datos se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

Los modelos de datos más habituales son:

- **Relacional (SGBDR):** representa a la base de datos como una colección de tablas. Estas bases de datos suelen utilizar SQL como lenguaje de consultas de alto nivel.



NORMALIZACIÓN DE BASES DE DATOS



Normalización

Se define

Como un proceso que transforma datos complejos a un conjunto de estructuras de datos

Beneficios

Consumo minimizado de espacio.
Menor repetición de datos.

Grados

Primera Forma Normal (1NF): Elimina columnas repetidas y se colocan en tablas separadas.

Segunda Forma Normal (2NF): Establece que toda columna que no es llave será dependiente de la llave primaria.

Tercera Forma Normal (3NF): Elimina la dependencia transitiva.

Primera Forma Normal:

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas.

Segunda Forma Normal:

La regla de la Segunda Forma Normal establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave primaria de la tabla para identificarlos.

Una vez alcanzado el nivel de la Segunda Forma Normal, se controlan la mayoría de los problemas de lógica. Podemos insertar un registro sin un exceso de datos en la mayoría de las tablas.

Tercera Forma Normal:

Una tabla está normalizada en esta forma si todas las columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas. Comentamos anteriormente que una dependencia transitiva es aquella en la cual existen columnas que no son llave que dependen de otras columnas que tampoco son llave.

Cuando las tablas están en la Tercera Forma Normal se previenen errores de lógica cuando se insertan o borran registros. Cada columna en una tabla está identificada de manera única por la llave primaria, y no deben haber datos repetidos. Esto provee un esquema limpio y elegante, que es fácil de trabajar y expandir.

Ejemplo: Ordenes de compra de clientes

ID_ORDEN	FECHA	ID_CLIENTE	NOM_CLIENTE	ESTADO	NUM_ITEM	DESC_ITEM	CANT	PRECIO
2301	2/23/03	101	MARTI	CA	3786	RED	3	35
2301	2/23/03	101	MARTI	CA	4011	RAQUETA	6	65
2301	2/23/03	101	MARTI	CA	9132	PAQ-3	8	4.75
2302	2/25/03	107	HERMAN	WI	5794	PAQ-6	4	5.0
2303	2/27/03	110	WE-SPORTS	MI	4011	RAQUETA	2	65
2303	2/27/03	110	WE-SPORTS	MI	3141	FUNDA	2	10

Primera Forma Normal – 1FN

Se eliminan columnas repetitivas.

- ORDENES

ID_ORDEN	FECHA	ID_CLIENTE	NOM_CLIENTE	ESTADO
2301	2/23/03	101	MARTI	CA
2302	2/25/03	107	HERMAN	WI
2303	2/27/03	110	WE-SPORTS	MI

- ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	DESC_ITEM	CANT	PRECIO
2301	3786	RED	3	35
2301	4011	RAQUETA	6	65
2301	9132	PAQ-3	8	4.75
2302	5794	PAQ-6	4	5.0
2303	4011	RAQUETA	2	65
2303	3141	FUNDA	2	10

Segunda Forma Normal – 2FN

Toda columna que no es clave es dependiente de la clave primaria.

La tabla ORDENES está en 2FN. Cualquier valor único de ID_ORDEN determina un sólo valor para cada columna. Por lo tanto, todas las columnas son dependientes de la llave primaria ID_ORDEN.

- ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	DESC_ITEM	CANT	PRECIO
2301	3786	RED	3	35
2301	4011	RAQUETA	6	65
2301	9132	PAQ-3	8	4.75
2302	5794	PAQ-6	4	5.0
2303	4011	RAQUETA	2	65
2303	3141	FUNDA	2	10

Por su parte, la tabla ARTICULOS_ORDENES no se encuentra en 2FN ya que las columnas PRECIO y DESC_ITEM son dependientes de NUM_ITEM, pero no son dependientes de ID_ORDEN. Lo que haremos a continuación es eliminar estas columnas de la tabla ARTICULOS_ORDENES y crear una tabla ARTICULOS con dichas columnas y la clave primaria de la que dependen.

- ARTICULOS_ORDENES

ID_ORDEN	NUM_ITEM	CANT
2301	3786	3
2301	4011	6
2301	9132	8
2302	5794	4
2303	4011	2
2303	3141	2

- ARTICULOS

NUM_ITEM	DESC_ITEM	PRECIO
3786	RED	35
4011	RAQUETA	65
9132	PAQ-3	4.75
5794	PAQ-6	5.0
4011	RAQUETA	65
3141	FUNDA	10

Tercera Forma Normal – 3FN

Eliminar dependencias transitivas. Tenemos que eliminar cualquier columna no clave que sea dependiente de otra columna no clave.

Los pasos a seguir son:

- Determinar las columnas que son dependientes de otra columna no llave.
- Eliminar esas columnas de la tabla base.
- Crear una segunda tabla con esas columnas y con la columna no llave de la cual son dependientes.

Al observar las tablas que hemos creado, nos damos cuenta que tanto la tabla ARTICULOS, como la tabla ARTICULOS_ORDENES se encuentran en 3FN. Sin embargo la tabla ORDENES no lo está, ya que NOM_CLIENTE y ESTADO son dependientes de ID_CLIENTE, y esta columna no es la llave primaria.

Para normalizar esta tabla, moveremos las columnas no llave y la columna llave de la cual dependen dentro de una nueva tabla CLIENTES. Las nuevas tablas CLIENTES y ORDENES se muestran a continuación.

- ORDENES

ID_ORDEN	FECHA	ID_CLIENTE
2301	2/23/03	101
2302	2/25/03	107
2303	2/27/03	110

- CLIENTES

ID_CLIENTE	NOM_CLIENTE	ESTADO
101	MARTI	CA
107	HERMAN	WI
110	WE-SPORTS	MI

En resumen:

La normalización es una técnica que se utiliza para crear relaciones lógicas apropiadas entre tablas de una base de datos. Ayuda a prevenir errores lógicos en la manipulación de datos. La normalización facilita también agregar nuevas columnas sin romper el esquema actual ni las relaciones.

Existen varios niveles de normalización: Primera Forma Normal, Segunda Forma Normal, Tercera Forma Normal, Forma Normal Boyce-Codd, Cuarta Forma Normal, Quinta Forma Normal o Forma Normal de Proyección-Unión, Forma Normal de Proyección-Unión Fuerte, Forma Normal de Proyección-Unión Extra Fuerte y Forma Normal de Clave de Dominio. Cada nuevo nivel o forma nos acerca más a hacer una base de datos verdaderamente relacional. Se discutieron las primeras tres formas. Éstas proveen suficiente nivel de normalización para cumplir con las necesidades de la mayoría de las bases de datos. Normalizar demasiado puede conducir a tener una base de datos ineficiente y hacer a su esquema demasiado complejo para trabajar. Un balance apropiado de sentido común y práctico puede ayudarnos a decidir cuándo normalizar.

TIPOS DE DATOS

Consideraciones previas

- ▶ Qué tipo de información se va a almacenar. Por ejemplo, no se pueden guardar caracteres en un campo cuyo tipo de datos sea numérico.
- ▶ El espacio de almacenamiento necesario (dimensionar el campo).
- ▶ Qué tipo de operaciones se van a realizar con los valores del campo. Pues, por ejemplo, no se puede calcular la suma de dos cadenas de texto.
- ▶ Si se desea ordenar o indexar por ese campo. Los criterios de ordenación difieren en función del tipo de dato, así, los números almacenados en un campo texto se ordenan según el valor de su código ASCII (1,10,11,2,20,...) que no coincide con la ordenación numérica.

Tipos de Datos (I)

- Numéricos Exactos:

Tipo	Desde	Hasta
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	$-10^{38} + 1$	$10^{38} - 1$
numeric	$-10^{38} + 1$	$10^{38} - 1$
money	-922,337,203,685,477.5808	+922,337,203,685,477.5807
smallmoney	-214,748.3648	+214,748.3647

Tipos de Datos (II)

- Numéricos Aproximados:

Tipo	Desde	Hasta
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

- Fechas / Horas:

Tipo	Desde	Hasta	Precisión
datetime	1 Enero de 1753	31 Diciembre 9999	3.33 ms.
smalldatetime	1 Enero de 1900	6 Junio 2079	1 minuto
time	N/A	N/A	1 ns.
date	1 Enero de 0001	31 Diciembre 9999	1 día
datetime2	1 Enero de 0001	31 Diciembre 9999	Variable
datetimeoffset	1 Enero de 0001	31 Diciembre 9999	Variable

Tipos de Datos (III)

- Texto

Tipo	Variable	Unicode	Capacidad
char	NO	NO	8000
varchar	SI	NO	8000
varchar(max)	SI	NO	2 ³¹
text	SI	NO	2,147,483,647
nchar	NO	SI	4000
nvarchar	SI	SI	4000
nvarchar(max)	SI	SI	2 ³⁰
ntext	SI	SI	1,073,741,823

Tipos de Datos (IV)

- Binarios

Tipo	Variable	Capacidad
binary	NO	8000
varbinary	SI	8000
varbinary(max)	SI	2 ³¹ FILESTREAM
image	SI	2,147,483,647

Tipos de Datos (V)

- Otros tipos de datos

Tipo	Comentario
XML	Almacena una instancia de XML
HierarchyID	Representa la posición en una jerarquía. No representa un árbol.
Table	Un tipo de datos especial que se utiliza para almacenar un conjunto de resultados para un proceso posterior.
Rowversion	Un número único para toda la base de datos que se actualiza cada vez que se actualiza una fila. (utilizar rowversion para versiones futuras)
Sql_variant	Un tipo de datos que almacena valores de varios tipos de datos aceptados en SQL Server, excepto text, ntext, rowversion y sql_variant
Uniqueidentifier	Un identificador exclusivo global (GUID), necesario para
Cursor	Una referencia a un cursor.

FIN
Gracias