

Capacitación SQL

Nivel inicial

Unidad 1/8

Índice general

1. Conceptos básicos de Base de Datos
2. Sistema de Gestión de Base de Datos
 - Base de datos relacionales
 - Relaciones
3. Tablas
4. Registros
5. Tipo de datos
 - Caracteres

1. Conceptos básicos de Base de Datos

En esta clase se presentan nociones de teoría de bases de datos que ayudarán a la comprensión de los conceptos utilizados a lo largo del curso.

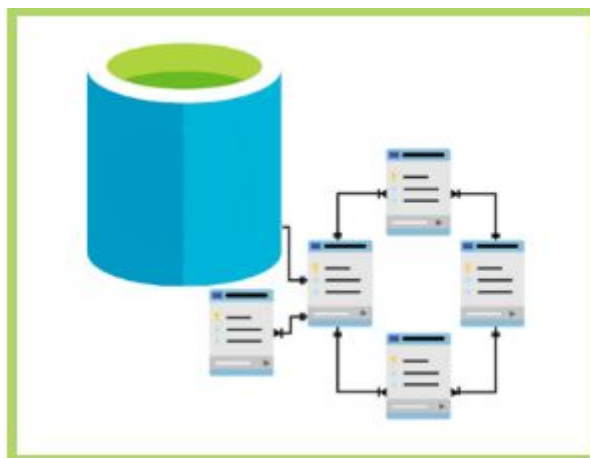
Una **base de datos**, como su nombre lo indica, es un conjunto de datos organizados y relacionados entre sí, recolectados, almacenados sistemáticamente y explotados para satisfacer las necesidades de información de una organización.

Desde el punto de vista informático, una base de datos está formada por el conjunto de datos almacenados y el sistema que permite su gestión, es decir su recolección, almacenamiento y explotación, el cual se denomina Sistema de **Gestión de Bases de Datos** (DBMS por sus siglas en inglés, DataBase Management System) o motor de base de datos.

2. Sistema de Gestión de Base de Datos

El objetivo principal es proporcionar una forma de almacenar y recuperar los datos de una base de datos, para generar información que sea práctica de acceder como eficiente.

Los sistemas de bases de datos se diseñan con el objetivo de manipular grandes volúmenes de datos. La gestión de éstos, implica tanto la definición de estructuras para el almacenamiento como la provisión de mecanismos para manipularlos. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, evitando posibles resultados anómalos durante la consulta.

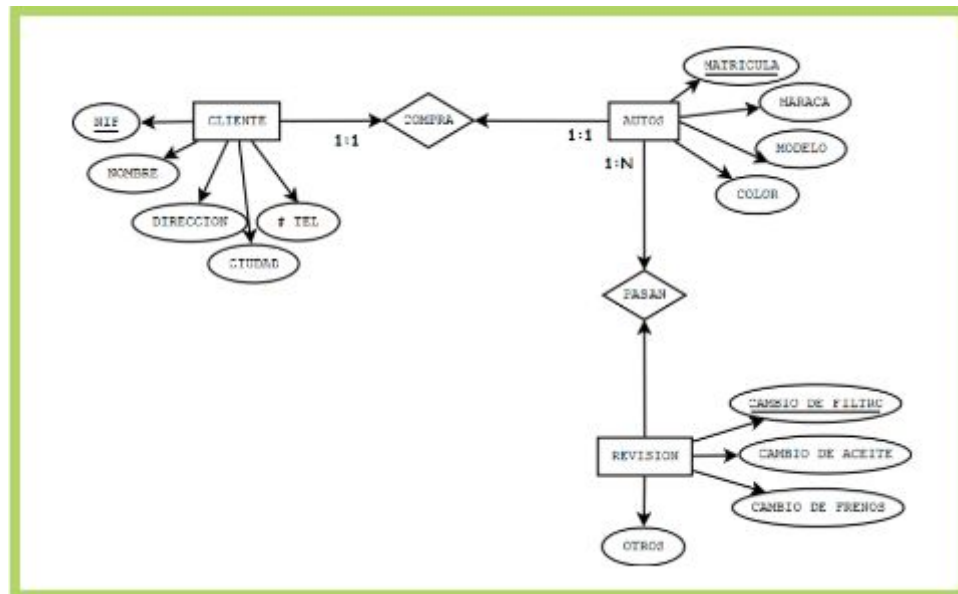


Base de datos relacionales

Bajo la estructura de la base de datos se encuentra el modelo de datos: una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia.

Existen muchas maneras de organizar y almacenar datos (como bases de datos jerárquicas, bases de datos orientadas a objetos, archivos, etc.). En este curso nos referiremos a **bases de datos relacionales**, en las cuales los datos se almacenan en tablas, las tablas están organizadas en campos, el contenido de los campos conforma registros y las tablas se relacionan entre sí, vinculándose por sus campos.

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real consistente en objetos básicos llamados **entidades** y de **relaciones** entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema que representa la estructura lógica completa de una base de datos. Constituyen la estructura o el modelo de la base de datos (esta estructura sigue un conjunto de reglas que hacen que la base de datos sea relacional).



Para manipular los datos se utiliza el lenguaje SQL (siglas en inglés: Structured Query Language), un lenguaje estructurado de consultas para bases de datos relacionales. SQL permite, además de consultar los datos, crear las estructuras para almacenarlos, insertarlos, modificarlos y eliminarlos, según sea necesario.

SQL es el lenguaje estándar de la mayoría de los sistemas gestores de bases de datos, aunque cada implementación suele incluir algunas funcionalidades o sintaxis particulares.

Relaciones

Una relación es una asociación entre diferentes entidades. Por ejemplo, se puede definir una relación que asocie al cliente López con el préstamo P-15. Esta relación especifica que López es un cliente con el préstamo número P-15.

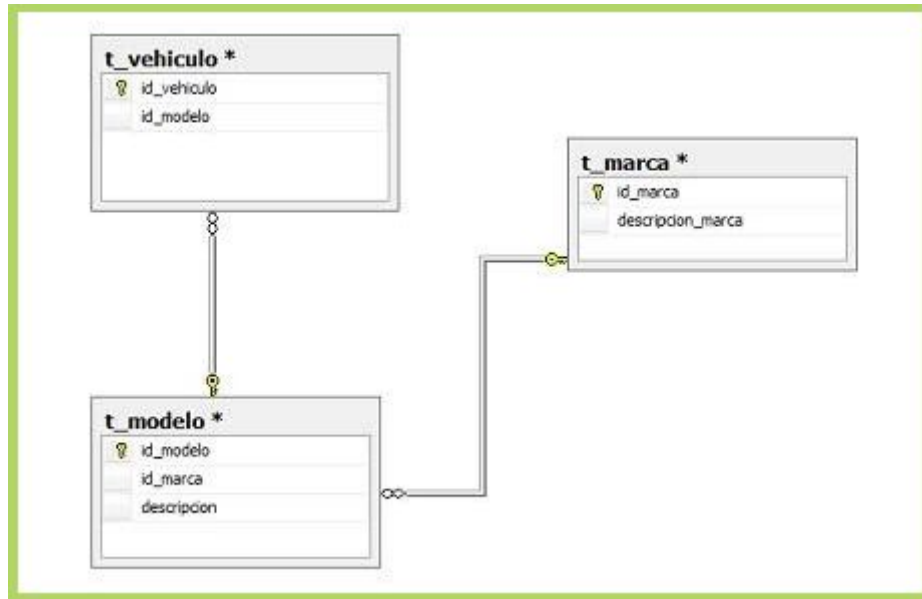
Las relaciones se establecen a partir de campos de las tablas. La definición de los campos que intervienen es parte fundamental del modelo de datos.

Las relaciones en una base de datos son el pilar fundamental, ya que permiten establecer las concordancias, asignaciones y relación entre las entidades o tablas, así mismo permite garantizar la integridad referencial de los datos, y además, evita la duplicidad de registros.

Para identificar de forma única cada registro de una tabla se define una **clave primaria**, (PK por sus siglas en inglés, Primary Key) con uno o más campos que contienen valores únicos y no nulos. Esta clave también corresponde al campo por el que otras tablas pueden referenciar los registros de esta tabla. Por ejemplo, Id_Cliente es la clave primaria de la tabla Clientes.

Para establecer una relación con otra tabla se definen claves foráneas (FK por sus siglas en inglés, Foreign Key). Esto implica que el contenido de este campo puede contener solamente los valores que existen como clave primaria en la tabla referenciada.

En el siguiente ejemplo, se observa la relación entre tablas. Aquellos campos con una llave a su izquierda, corresponden a claves primarias.

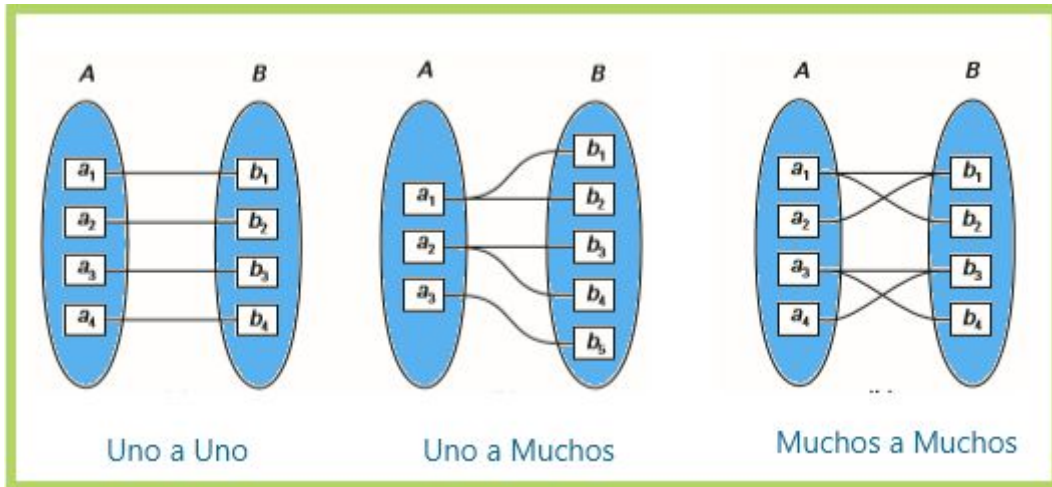


Tipos de relaciones

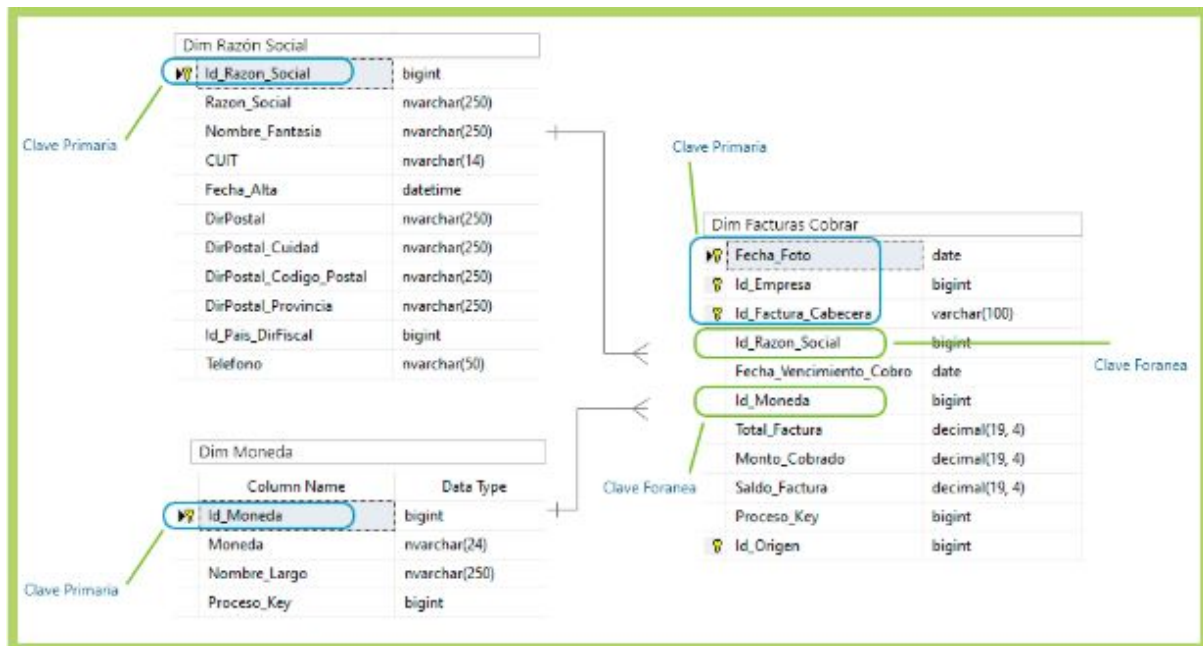
- **Uno a Uno:** En este tipo de relaciones se establecen entre un registro de una tabla y otro registro de una segunda tabla. Un ejemplo uno a uno sería presidente y País, ya que un solo presidente puede liderar un solo país y a su vez un país solo puede contar con un presidente.

Nota: En este tipo de relación se enlazan los campos establecidos como Primary Key de ambas tablas, normalmente no aportan mucho al mejoramiento del diseño de la base de datos pero si a la organización y fácil acceso de la información.

- **Uno a muchos:** Se establecen entre varios registros de una tabla y un registro de otra tabla. Por Ejemplo, una tabla de Clientes con Pedidos, en donde la relación sería entre clientes y pedidos, donde un cliente puede realizar muchos pedidos.
- **Muchos a muchos:** Esta relación se produce cuando varios registros de una tabla se asocian a varios registros de otra tabla. Por ejemplo, una relación de muchos a muchos puede ser entre los clientes y los productos: los clientes pueden comprar varios productos y los productos pueden ser comprados por muchos clientes.



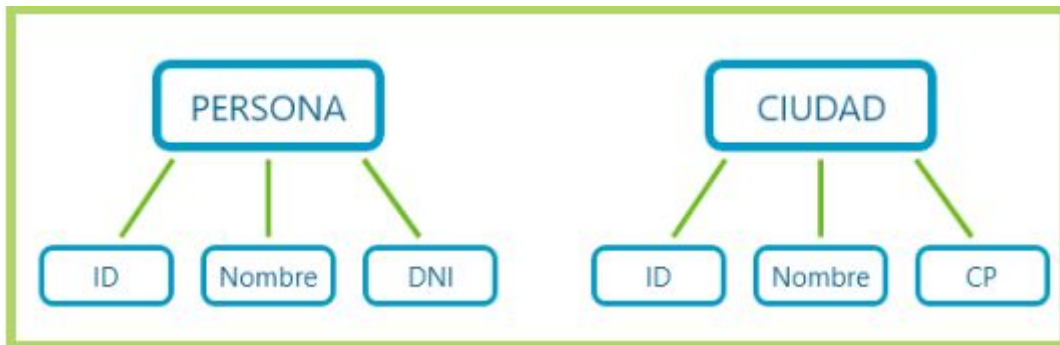
En la siguiente imagen se observa un diagrama de entidad relación donde la tabla **"Dim Facturas Cobrar"** tiene como clave foránea **"Id_Razon_Social"** con una cardinalidad de uno a muchos, esto quiere decir que una Razón social puede estar muchas veces en la Tabla Dim Facturas Cobrar, y que cada Factura va a tener asociado una Razón Social.



3. Tablas

Una tabla/entidad es una "cosa" u "objeto" en el mundo real que es distinguible de todos los demás objetos, en ella se guardan los datos recogidos por un programa. Puede ser concreta, como una persona o un libro, o puede ser abstracta, como un préstamo, unas vacaciones o un concepto.

Una entidad tiene un conjunto de propiedades, y los valores para algún conjunto de propiedades pueden identificar una entidad de forma unívoca. Por ejemplo, el D.N.I. 67.789.901 identifica unívocamente una persona particular.



Su estructura se asemeja a la vista general de un programa de hoja de cálculo.

Las tablas se componen por **Campos / Atributos**: Corresponde al nombre de cada columna.

Debe ser único y además tener un tipo de dato asociado.

Posibles atributos pueden ser id-cliente, nombre-cliente, calle-cliente y ciudad-cliente.

Un conjunto de datos de campos forman un registro.

	Column Name	Data Type	Allow Nulls
Nombre del Campo	Id_Cliente	bigint	<input type="checkbox"/>
	Id_ClienteServicios	nvarchar(6)	<input checked="" type="checkbox"/>
	Alianza_Cliente	nvarchar(125)	<input checked="" type="checkbox"/>
	Alianza	nvarchar(125)	<input checked="" type="checkbox"/>
	Cliente	nvarchar(125)	<input checked="" type="checkbox"/>
	Nombre_Carpeta	nvarchar(100)	<input checked="" type="checkbox"/>
	CodigoPostal	nvarchar(4)	<input checked="" type="checkbox"/>
	Localidad	nvarchar(125)	<input checked="" type="checkbox"/>
	Provincia	nvarchar(50)	<input checked="" type="checkbox"/>
	Pais	nvarchar(3)	<input checked="" type="checkbox"/>
	proceso_key	bigint	<input checked="" type="checkbox"/>

4. Registros

Corresponde a cada fila que compone la tabla. Supone un conjunto de datos que mantienen una vinculación entre sí. La totalidad de las filas de una tabla respeta una estructura idéntica, una característica que permite manipular y realizar cálculos con la información.

El empleo de registros tiene el fin de almacenar información y datos, generar una relación y colocarla al alcance bajo un índice o sistema de orden que permita su acceso y uso en cualquier momento. Los registros son el método que tanto el usuario como el sistema informático utilizan para acceder y utilizar toda la información.

		CAMPO						
	ProductKey	ProductAlternateKey	ProductSubcategoryKey	WeightUnitMeasureCode	SizeUnitMeasureCode	EnglishProductName	SpanishProductName	FrenchProductName
1	1	AR-5381	NULL	NULL	NULL	Adjustable Race		
2	2	BA-8327	NULL	NULL	NULL	Bearing Ball		
3	3	BE-2349	NULL	NULL	NULL	BB Ball Bearing		
4	4	BE-2908	NULL	NULL	NULL	Headset Ball Bearings		
5	5	BL-2036	NULL	NULL	NULL	Blade		
6	6	CA-5965	NULL	NULL	NULL	LL Crankarm		
7	7	CA-6738	NULL	NULL	NULL	ML Crankarm		
8	8	CA-7457	NULL	NULL	NULL	HL Crankarm		
9	9	CB-2903	NULL	NULL	NULL	Chaining Bolts		
10	10	CN-6137	NULL	NULL	NULL	Chaining Nut		

TABLA

5. Tipo de datos

Los campos guardan un tipo de dato previamente definido que pueden ser, por ejemplo, alfabéticos, numéricos, lógicos (guardan verdadero o falso) o de fecha. Los tipos de datos hacen referencia al tipo de información que se manipula, donde la unidad mínima de almacenamiento es el dato, también se puede considerar como el rango de valores que puede tomar una variable durante la ejecución del programa.

Los nombres específicos para los tipos de datos dependen del gestor de base de datos que se utilice.

Caracteres

El tipo de dato carácter es un dígito individual el cual se puede representar como numéricos (0 al 9), letras (a-z) y símbolos (!"#\$%&'\.).}

Caracteres numéricos

Este tipo de dato puede ser real o entero, dependiendo del tipo de dato que se vaya a utilizar.

Caracteres Booleanos

Este tipo de dato se emplea para valores lógicos, se puede definir como datos comparativos dicha comparación devuelve resultados lógicos (Verdadero o Falso).

También existen tipo de datos como Fecha/Hora, Geométricos, Direcciones de red.

A continuación, se detallan algunos de los tipos de datos:

Tipo de dato		Descripción
Numérico	Smallint	Entero con valores entre -32768 y +32767
	Integer	Entero con valores entre -2147483648 y +2147483647
	Bigint	Entero con valores entre -9223372036854775808 y +9223372036854775807
	Real	Número decimal (hasta 6 dígitos decimales)
	double precision	Número decimal (hasta 15 dígitos decimales)
	Serial	Entero autoincremental
Alfabético	character (n), char (n)	Cadena de caracteres de tamaño fijo
	character varying (n), varchar (n)	Cadena de caracteres de tamaño variable
	Text	Cadena de caracteres de tamaño ilimitado
Lógico	Boolean	Valor lógico o booleano (true/false)
Fecha	Date	Fecha (sin hora)
	Timestamp with time zone	Fecha y hora con zona horaria
	Timestamp without time zone	Fecha y hora sin zona horaria

Algunos ejemplos de utilización de los tipos de datos más comunes:

Tipo de dato		Ejemplo Campo	Ejemplo Registro
Numérico	Integer	Edad	27
	Real	Altura	1,58
	Serial	ID_Cliente	350
Alfabético	character (n), char (n)	Patente	ABC123
	character varying (n), varchar (n)	Nombre	María
	Text	Comentarios	Esto es un comentario
Lógico	Boolean	Activo	True/False
Fecha	Date	Inicio_Proyecto	04/04/2020

----- *Seguí reforzando tu conocimiento en la próxima unidad del programa.