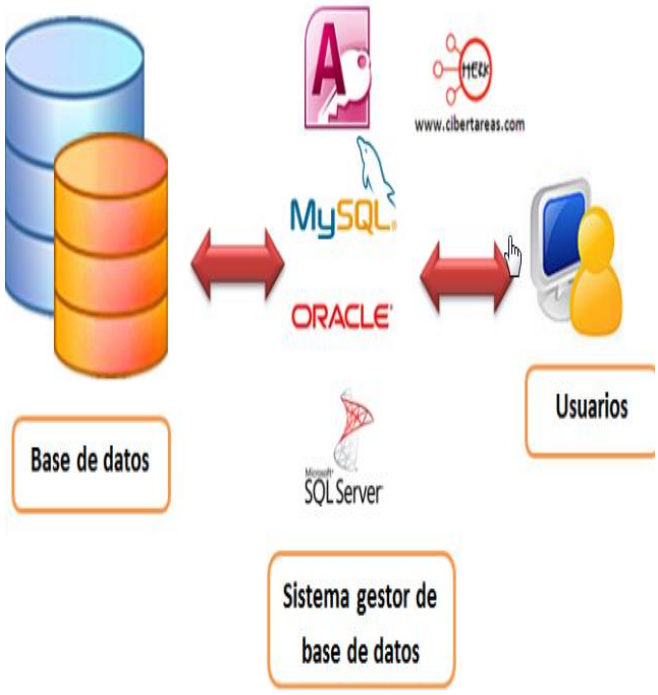
INSTITUCION EDUCATIVA TECNICO "DOMINGO BELISARIO GOMEZ"
Autorizada según Resolución N°. 0480 26 abril de 2004
DANE 119100002474 Nit. 891 501 59-07
Pre-escolar, Primaria, Básica Secundaria y Media Técnica.
Especialidades: Sistemas, Comercio e Industria: Ebanistería y Metalistería
Carrera 3 Nª 11-10 Celular 313 601 05 97 Barrio San Francisco Bolívar – Cauca

PRESENTACIÓN. Estimado aprendiz muy buen día. En esta guía de aprendizaje se le invita a participar y desarrollar las diferentes actividades propuestas, que tienen un propósito fundamental y es conocer el diseño de las base de datos, cómo crear un modelo entidad relación y como sistematizar los procesos manuales de la información en registros digitales y poder realizar consultas, ingresos, actualización y eliminación de los datos. Tendrá la oportunidad de desarrollar sus capacidades de pensamiento, habilidades y tener una mayor destreza para resolver problemas del entorno productivo y social, donde se insertará como un futuro trabajador, un mejor ciudadano y una mejor persona, de ahí la importancia para el **SENA** de lograr una Formación Profesional Integral.

EMPECEMOS

¿PERO CÓMO SURGIERON LAS BASES DE DATOS?

Las bases de Datos surgen de los inconvenientes que se estaban presentado en los sistemas de archivo; en donde estos, almacenaban los registros en varios archivos, se tenía que desarrollar aplicaciones especiales para extraer estos registros y añadirlos a la aplicación adecuada para la información deseada; esto inconvenientes son:



- **Redundancia e inconsistencia en los datos:** Debido a que los archivos y los programas de aplicación eran creados por diferentes programadores, formatos y lenguajes; se podía presentar que la información se encontrara duplicada e inconsistente.

- **Dificultad de Acceso:** Este inconveniente se presentaba cuando se requerían consultas muy específicas sobre los registros y la aplicación solo generaba un informe general. La persona tenía que extraer su consulta de manera manual con una serie de datos que eran irrelevantes, innecesarios o confidenciales. o decirle al programador que realizara una aplicación para esta consulta, la cual podría demorar demasiado tiempo.

- **Aislamiento de datos:** Debido a que los registros estaban en diversas aplicaciones y formatos, era muy complicado programar nuevas aplicaciones que permitieran la recuperación de datos o archivos apropiados.

- **Problemas Integridad:** Muchas aplicaciones tiene datos que deben tener ciertas reglas o estar bajo ciertos parámetros, como por ejemplo que una cuenta de ahorros no este negativa, que la fecha de aprobación de un crédito no menor a la de nacimiento, etc. El inconveniente surgía cuando se presentaban nuevas restricciones ya que era muy complicado que este la cumpliera o peor cuando estas implicaban diferentes elementos de datos de diferentes archivos.

- **Problemas de Atomicidad:** Todo sistema informático o de cómputo no está exento de fallas, pero cuando se presenta una es muy importante que los datos se restablezcan al estado de consistencia antes de ocurrir el evento. Un ejemplo es cuando se realizaba un transacción bancaria y se caía el sistema, no se sabía si la transacción se hacía o no; algunas veces se tenía que realizar procesos de cuadre manual para responderle

al cliente.

- **Anomalías de acceso concurrente:** Esto surge debido a que muchos usuarios tiene acceso al sistema de archivos y realizan actualizaciones las cuales pueden crear inconsistencias, ya que no hay un control de entrada a la aplicación.

- **Problemas de Seguridad:** Este inconveniente se presenta cuando no se tienen claramente definidos los permisos de los usuarios y los usuarios pueden ver información confidencial, o cuando estas restricciones eran específicas y no se generalizaba sobre todo el sistema.

Ejemplos de base de datos:

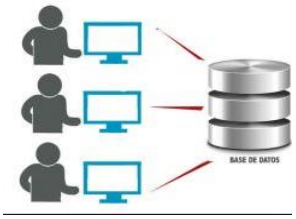
La guía telefónica



Las fichas de una biblioteca



Una agenda con los datos de nuestros amigos



SISTEMAS DE GESTION DE BASE DE DATOS (SGBD)



Un **gestor de base de datos** es conocido como un conjunto de programas que sirven de interfaz entre el usuario y la base de datos para tener una sencilla manipulación de los datos. Un **SGBD** debe permitir especificar tipos y estructuras, permite la manipulación de los datos mediante consultas y la actualización de la **base de datos** de manera sencilla, existen diferentes gestores de bases de datos, aquí te mostrare los mas usados.

MySQL



Es uno de los gestores de **bases de datos** más usados tanto por la comunidad estudiantil como por las empresas, está desarrollada bajo las licencias de GPL y la licencia comercial de Oracle. Nació en el año de 1995, de la mano de **Michael Widenius y David Axmark**, surgió de la necesidad de un buen sistema de almacenamiento, fue así como salió **MySQL** y junto con ello **MySQL AB**. Este gestor de base de datos es el más popular de código abierto y es utilizada por **Twitter, Facebook y YouTube** gracias a su rendimiento y confiabilidad.

ORACLE



Es conocido como uno de los **gestores de BD** más completos gracias a su estabilidad y su soporte multiplataforma, el costo de este depende del tipo de licencia que se adquiera y se puede usar en distintos sistemas operativos, una de las desventajas a parte del costo es la vulnerabilidad en su seguridad, al igual que **MySQL** es uno de los más usados a nivel mundial.

SQLite



Es un **sistema gestor de base de datos** compatible con **ACID** (Atomicidad, Consistencia, Aislamiento y Durabilidad), **ACID** son las características que clasifican las transiciones en los **SGBD**,

cuenta con una pequeña biblioteca creada en C, a diferencia de los demás que se funcionan como un proceso aparte este se integra al mismo sistema.

SQL Server

Este **SGBD** está enfocado para entornos empresariales, originalmente el código de **MySQL server** fue desarrollado por **Sybase** y durante un tiempo **Microsoft** y **Sybase** siguieron generando productos de **MySQL server**, tiempo después Sybase cambio el nombre de su producto llamándolo **Adaptive Server Enterprise.**

SQL server cuenta con un entrono grafico para administración, en este se pueden usar comandos DDL y DML de manera gráfica, es escalable, estable y permite administrar información de otros servidores.

MICROSOFT ACCESS

Este gestor de base de datos viene incluido dentro de la paquetería de **Microsoft office**, es de fácil uso permite crear **bases de datos** rápidamente, cuenta con plantillas para crear aplicaciones sencillas pero funcionales, su lanzamiento fue en el año de 1992.

PostgreSQL

Es un **gestor de base de datos** bajo la licencia, sustituye el uso de multihilos por multiprocesos garantizando de esta manera la estabilidad del sistema, es bastante popular y fácil de administrar, la sintaxis de **SQL** que usa es bastante rápida y fácil de aprender, es multiplataforma, en comparación con otros gestores es lento en actualizaciones y su consumo de recursos es más alto que el de **MySQL**.

El objetivo principal de estos gestores es hacer más amable la interacción con la base de datos, existen un sinfín de ellos, cada uno con características distintas algunas de ellos completamente.

FUNDAMENTOS Y COMPONENTES ACCESS



Una vez que determinamos los inconvenientes que presentaban los Sistemas de Archivos, surge el Sistema Gestor de Bases de Datos (SGBD), el cual consiste en consiste en un conjunto de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

Microsoft ACCESS es un Sistema gestor de Base de Datos Relacional, cuyo objetivo es proporcionar una manera de almacenar y recuperar la información de tal forma que sea practica y eficiente. El término **relacional** se da por la metodología utilizada en la interacción entre sus datos a través de relaciones, que explicare mas adelante.

Una base de datos es un conjunto de datos relacionados entre sí, a través de un **SGBD** que facilita el almacenamiento, gestión y recuperación de datos.

Componentes de la base de Datos relacional. Como Access en SGBD relacional, se explicaran los principales términos:

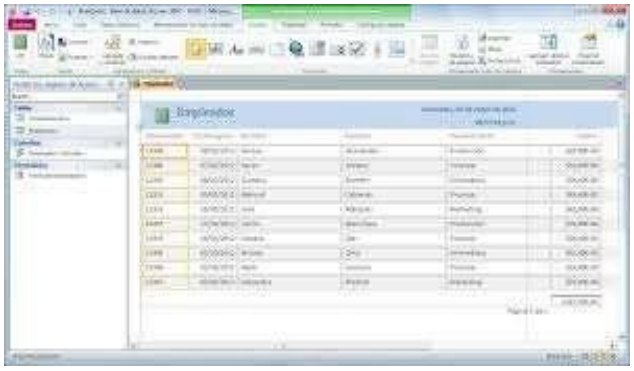
Tablas: Son objetos que se utilizan para almacenar datos referentes aun tema particular; Por ejemplo Clientes, Proveedores, Servicios, Alumnos, etc._Una tabla esta definida por **campos (columnas)** donde cada uno de ellos tiene un tipo de datos especifico como por ejemplo de tipo numero, texto, fecha, etc._Los **registros (Filas)** es donde se almacena la información individual de cada tabla, consta de al menos un campo.

INSTITUCION EDUCATIVA TECNICO "DOMINGO BELISARIO GOMEZ"
Autorizada según Resolución N°. 0480 26 abril de 2004
DANE 119100002474 Nit. 891 501 59-07
Pre-escolar, Primaria, Básica Secundaria y Media Técnica.
Especialidades: Sistemas, Comercio e Industria: Ebanistería y Metalistería
Carrera 3 Nª 11-10 Celular 313 601 05 97 Barrio San Francisco Bolívar – Cauca



Formularios: Son una interfaz que permite ingreso y modificación de los datos de una manera mas agradable a través de un ambiente gráfico, el cual puede ser personalizado o incluir botones de trabajo. Tienen la ventaja de: mostrar datos de varias tablas, calcular campos, crear y utilizar gráficos, utilizar fuentes, colores y títulos personalizados.

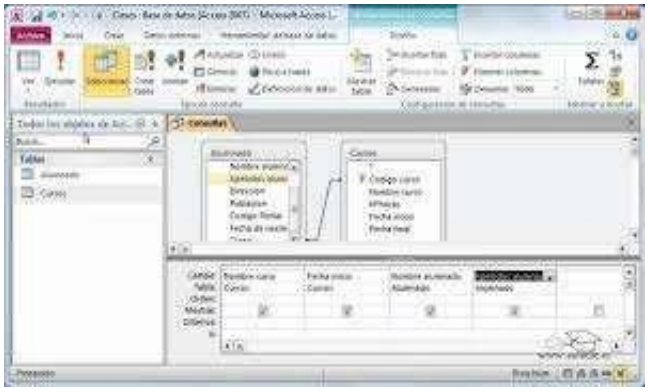




Informes: Es la forma de presentar los datos contenidos en una o varias tablas para un formato impreso o visual, normalmente los informes son los resultados de algún tipo de consulta. Tienen ventajas similares a los formularios ya que se pueden personalizar de una manera profesional.

Consultas: Son las que hacen el trabajo en la base de datos, cuya función es recuperar datos específicos de diferentes tablas, las cuales están unidas a través de relaciones. Básicamente las consultas son las preguntas que le hacemos a la Base de Datos

Existen varios criterios de como son: Consultas de selección, creación de tabla, crear registros, de referencias cruzadas, de eliminación.



RELACIONES ENTRE TABLAS

Access es una de las herramientas más potentes para el manejo de datos porque, sin necesidad de procedimientos complejos, permite relacionarlos para obtener de ellos la información deseada en el momento en que la requerimos. Por eso, aprenderemos a crear relaciones entre tablas y comprenderemos la importancia de las claves principales. Además, conoceremos los distintos tipos de relaciones.

Clave Principal

Para crear relaciones entre tablas, debemos indicar un campo especial en cada una de ellas, que se denomina **clave principal**.

Qué es una clave principal. En el proceso de elaboración de tablas, siempre debemos incluir un campo que nos permita identificar cada registro de manera única e irrepetible. A continuación, en la **Tabla**, mostramos ejemplos de campos que identifican registros para distintas tablas.

EJEMPLOS DE CAMPOS CLAVE

TABLA	CAMPO QUE IDENTIFICA CADA REGISTRO
Productos	Código_artículo
Empleados	ID Empleado
Sucursales	Nº Sucursal

La clave principal nos permite identificar de manera única cada registro.

Este método es fundamental para localizar e identificar un dato, ya que permite hacer referencia a él sin nombrar cada uno de sus atributos. Por lo tanto, es más simple mencionar al empleado **1**, que decir **Juan Pérez**, **26 años**, **sexo masculino**, **número de documento**, etcétera. El campo que se emplea para identificar cada registro con un valor único se denomina **campo clave** o **clave principal**.
A su vez, el campo clave de una tabla es frecuentemente utilizado en otra, con el fin de poder relaciona entre sí. A estos campos los llamamos **clave externa**.

Ventas

Fecha de la venta	Nro de sucursal	Id empleado	Código_artículo	Cantidad
03/06/2013	2	3	AC432D	10
04/06/2013	3	5	AC431D	5
04/06/2013	4	6	AC433D	4
04/06/2013	2	3	AC434D	6
05/06/2013	1	2	AC431D	8
05/06/2013	3	1	AC432D	3
05/06/2013	4	3	AC435D	4
06/06/2013	1	4	AC433D	5
07/06/2013	2	2	AC432D	15
08/06/2013	3	3	AC435D	2
09/06/2013	2	1	AC431D	4
09/06/2013	4	2	AC434D	7
10/06/2013	2	6	AC435D	6
10/06/2013	3	5	AC435D	8
11/06/2013	2	4	AC435D	4
11/06/2013	4	3	AC431D	6
12/06/2013	2	2	AC434D	3
12/06/2013	3	1	AC434D	8

Los campos clave externos son campos clave de otras tablas que se utilizan como dato complementario. En este caso, **Nro de sucursal**, **Id empleado** y **Código_artículo**. Por su lado, que un campo sea **clave principal** implica que:

El campo queda automáticamente indexado sin duplicados: si ingresamos un valor repetido en el campo clave, Access nos muestra un mensaje de error. Entonces, debemos pulsar el botón **Aceptar** y corregir el dato duplicado.
No es posible omitir valores en el campo: todos los registros deberán contener un valor en el campo clave, de lo contrario, Access muestra un mensaje de error. Para corregirlo, debemos pulsar el botón **Aceptar** y, a continuación, ingresar los datos en las celdas vacías del campo clave.

El campo clave no puede tener valores repetidos, ni tampoco estar vacío.

Las herramientas para relacionar tablas nos permiten acceder rápidamente a la información almacenada, sin necesidad de abrir cada una de las tablas que contienen los datos. Por lo tanto, al proceso de **relación** podemos definirlo como: la vinculación entre dos tablas a través de un campo en común. A continuación, veremos cómo es posible crear la relación entre tablas. Es importante tener en cuenta que, antes de eliminar o modificar la clave principal de un campo, debemos verificar que esta no participe de ninguna relación entre tablas. Si se da este caso, Access mostrará un mensaje de error que indicará que no es posible eliminar o modificar dicha clave.

Tipos de relaciones

Si bien es cierto que cada tabla almacena información independiente, esto no quiere decir que las relaciones entre sus datos no puedan darse de distintas formas. Por ejemplo, puede darse desde un solo registro y terminar mostrando varios resultados compatibles. Esto lo podemos ver si relacionamos, por ejemplo, una tabla **Empleados** (que almacena los datos personales de cada empleado) y una tabla **Ventas** (que registra las ventas realizadas, más el empleado que las efectúa) porque, al abrir la primera, podremos ver el listado de ventas que realizó cada uno de ellos.

Debemos tener en cuenta que para crear una relación entre tablas, los campos que se van a relacionar tienen que poseer el mismo tipo de datos o ser compatibles (por ejemplo, **Número** con **Autonumérico**).

Los tipos de relaciones que se pueden establecer entre tablas son:
Uno a varios: los datos de un campo pueden tener múltiples registros en el campo de la otra tabla. Por ejemplo, un empleado puede tener registradas varias ventas, por lo tanto, el campo **Id Empleado** de la tabla **Empleados** posee relación uno a varios registros con respecto al campo **Id Empleados** de la tabla **Ventas**.

Varios a varios: los datos de un campo pueden tener varios registros en la otra tabla y viceversa. Por ejemplo, un artículo puede incluir varias ventas y, a su vez, varias ventas pueden referirse a un mismo producto.

Uno a uno: en este tipo de relación, a cada registro de una tabla le corresponde un solo registro en la otra. Por ejemplo, el campo **Id Empleados** de la tabla **Empleados** tiene solamente un registro que le corresponde en el campo **Id Empleado** de una tabla que refleje el total de ventas de cada empleado, ya que no es posible que un mismo empleado tenga dos totales por sus ventas

¿Qué es SQL?

SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. Los programas de bases de datos relacionales, como Microsoft Office Access, usan SQL para trabajar con datos.
SQL se usa para describir conjuntos de datos que pueden ayudarle a responder preguntas. Al usar SQL, debe usar la sintaxis correcta. La sintaxis es el conjunto de reglas mediante las que se combinan correctamente los elementos de un idioma. La sintaxis SQL se basa en la sintaxis del idioma inglés y usa muchos de los mismos elementos que la sintaxis de Visual Basic para Aplicaciones (VBA).

Por ejemplo, una simple instrucción SQL que recupera una lista de los apellidos de contactos cuyo nombre es María podría ser similar a esto:

Por ejemplo, una simple instrucción SQL que recupera una lista de los apellidos de contactos cuyo nombre es María podría ser similar a esto:

```
SELECT Last_Nam
FROM Cliente
WHERE Nombre=”María”
```

Instrucciones SELECT

Para describir un conjunto de datos con SQL, se escribe una instrucción SELECT. Una instrucción SELECT contiene una descripción completa de un conjunto de datos que quiere obtener de una base de datos. Se incluye lo siguiente:

- Qué tablas contienen los datos.
-Cómo se relacionan los datos de orígenes diferentes.
-Qué campos o cálculos proporcionarán los datos.
-Criterios que los datos deben cumplir para ser incluidos.
-Si se deben ordenar los datos y, en caso de ser así, cómo deben ordenarse.

Cláusulas SQL

Como una frase, una instrucción SQL tiene cláusulas. Cada cláusula realiza una función de la instrucción SQL. Algunas cláusulas son necesarias en una instrucción SELECT. En la siguiente tabla, se enumeran las cláusulas SQL más comunes.

Cláusula SQL	Función	Obligatorio
SELECT	Muestra una lista de los campos que contienen datos de interés.	Sí.
FROM	Muestra las tablas que contienen los campos de la cláusula SELECT.	Sí.
WHERE	Especifica los criterios de campo que cada registro debe cumplir para poder ser incluido en los resultados.	No.
ORDER BY	Especifia la forma de ordenar los resultados.	No.
GROUP BY	En una instrucción SQL que contiene funciones de agregado, muestra los campos que no se resumen en la cláusula SELECT.	Solo si estan estos campos.
HAVING	En una instrucción SQL que contiene funciones de agregado, especifica las condiciones que se aplican a los campos que se resumen en la instrucción SELECT.	No.

¿Qué es el modelo entidad-relación?

Este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de BBDD (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos. Además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.

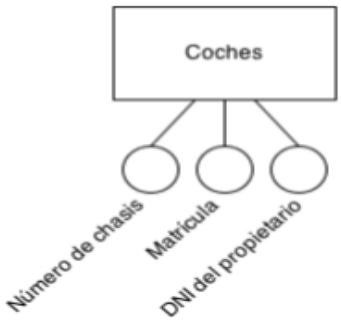
Elementos del modelo entidad-relación

Entidad. Representan cosas u objetos (ya sean reales o abstractos), que se diferencian claramente entre sí.
Para poder seguir un ejemplo trabajaremos sobre un taller mecánico, donde se podría crear las siguientes entidades:
Coches (objeto físico): contiene la información de cada taller.
Empleado (objeto físico): información de los trabajadores.
Cargo del empleado (cosa abstracta): información de la función del empleado.

Estas entidades se representan en un diagrama con unos rectángulos, como los siguientes.



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. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha.).
Siguiendo el ejemplo de antes podemos analizar los atributos de nuestra entidad "Coches", que nos darán información sobre los coches de nuestro supuesto taller.
Unos posibles atributos serían los siguientes: número de chasis, matrícula, DNI del propietario, marca, modelo y muchos otros que complementen la información de cada coche.
Los atributos se representan como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos, como a continuación.



En un modelo relacional (ya implementado en una base de datos) una ejemplo de tabla dentro de una BBDD podría ser el siguiente.

Número de chasis	Matrícula	DNI del propietario
5tfem5f10ax007210	4817 BFK	45338600L
6hsen2j98as001982	8810 CLM	02405068K
5rgsb7a19js001982	0019 GGL	40588860J

Este ejemplo es con tres atributos, pero un coche podría tener cientos (si fuese necesario) y seguirían la misma estructura de columnas, tras implementarlo en una BBDD.

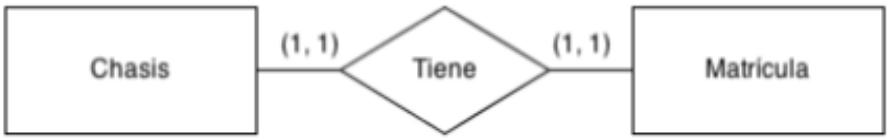
Relación. Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.
Por ejemplo, los empleados del taller (de la entidad "Empleados") tienen un cargo (según la entidad "Cargo del empleado"). Es decir, un atributo de la entidad "Empleados" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "Cargo del empleado".
Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.



Relaciones de cardinalidad

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades. Es decir, en el caso anterior cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.
Esto complementa a las representaciones de las relaciones, mediante un intervalo en cada extremo de la relación que especifica cuantos objetos o cosas (de cada entidad) pueden intervenir en esa relación.

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).



Uno a varios o varios a uno: Determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.



Varios a varios: Determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.



Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una "n").

Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad) o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:
Superclave: Aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

Clave primaria: Identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

Clave externa o clave foránea: Este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave. Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error.

Fases para crear una base de datos de inicio

A nivel teórico, los pasos a seguir a la hora de crear una base de datos son los siguientes:

1. Recopilación de toda la información del cliente para saber sus necesidades. Es una parte muy importante, porque al fin y al cabo es quién solicita nuestros servicios. Se deben realizar tantas entrevistas como sean necesarias.
2. Diseño del modelo entidad-relación (modelo conceptual). Consiste en plasmar en un diagrama las entidades, atributos y relaciones definidos en los requerimientos recogidos en la primera fase.
3. Transformación de este diagrama en el modelo relacional, basado en relaciones y tablas.
4. Toma de decisión sobre qué Sistema de Gestión de Bases de Datos (SGBD) utilizaremos. Algunos de los más utilizados son Oracle y MySQL.
5. Implementación del modelo relacional en el programa que gestiona la base de datos, es decir, el modelo físico.

Elementos del modelo entidad-relación

El diseño del modelo entidad-relación, corresponde a la segunda fase de la creación de una base de datos, que se realiza una vez hemos recopilado toda la información por parte del cliente. Los elementos básicos de un modelo entidad-relación son las entidades, las relaciones, los atributos y las cardinalidades.



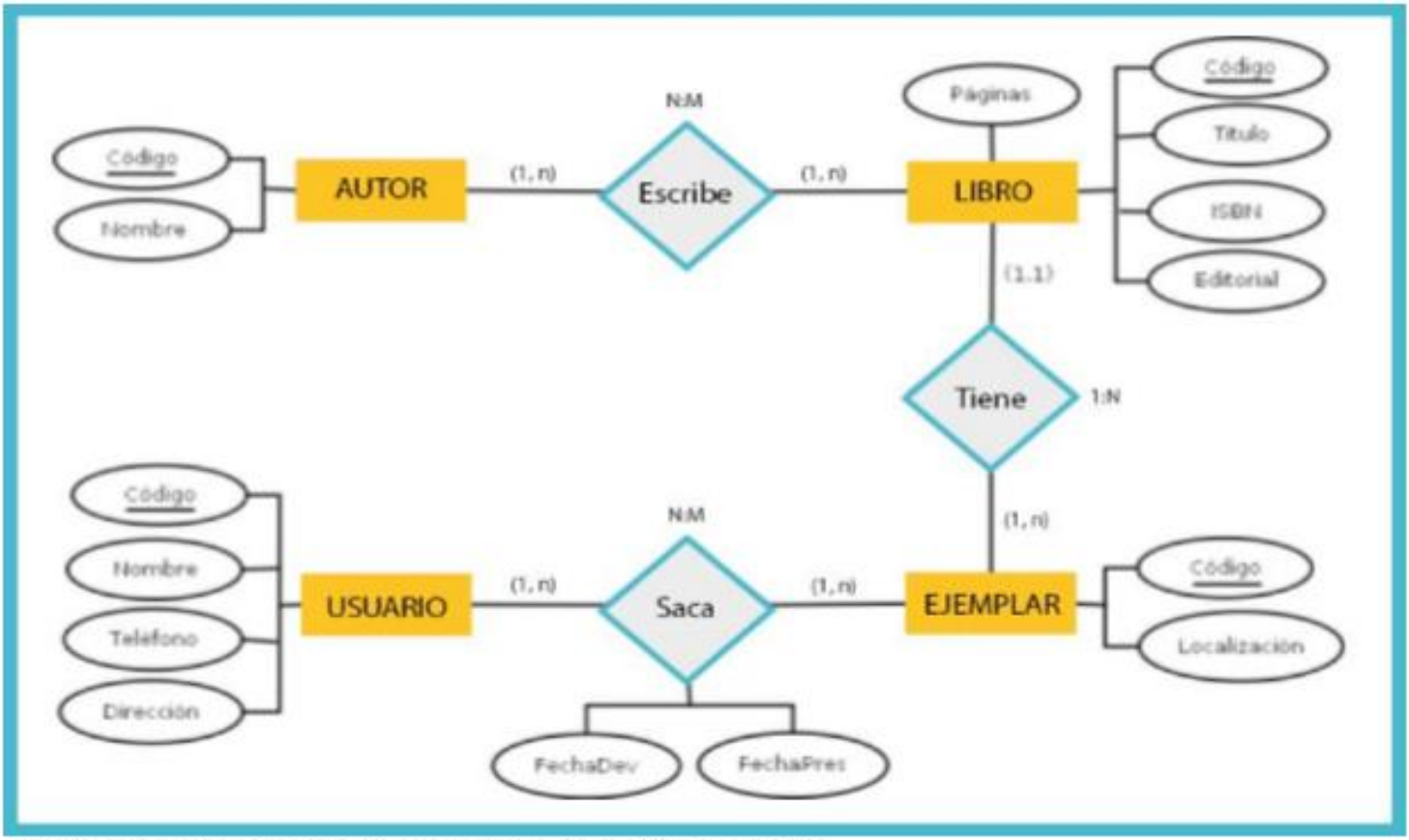
Esquema simple de una relación entre dos entidades y sus cardinalidades.

-Las diferentes entidades (objetos o sujetos de la base de datos de los que queremos almacenar información) tienen atributos -es decir, propiedades y características-, y las entidades se relacionan entre ellas. También se pueden representar entidades débiles, cuya existencia está ligada a otra entidad (fuerte).

-Un atributo o un conjunto de atributos de una entidad puede ser clave primaria, entendida como una propiedad que identifica de forma única un registro de una tabla. Por ejemplo, el DNI en el registro de un alumno, del que también guardamos otra información -atributos- como el nombre, los apellidos, el teléfono de contacto y la carrera en la cual se ha matriculado.

-Por último, se indica la cardinalidad, consistente en la participación que hay entre las entidades.

Ejemplo de entidad-relación. Aquí vemos un ejemplo de cómo quedaría plasmado gráficamente esta base de datos en que las entidades son autor, libro, ejemplar y usuario.



Ejemplo de la representación de un modelo entidad-relación.

Como vemos en el diagrama, las entidades se ilustran con un rectángulo (en el caso que hubiera una entidad débil, se representa con un doble rectángulo). Por su parte, las relaciones suelen ser un verbo, se representan con un rombo y vinculan las entidades, mientras que en óvalos se representan los atributos y se subrayan aquellos que son clave primaria (en el ejemplo, las claves primarias son los códigos de cada entidad).
Respecto a las cardinalidades, para poner un ejemplo ilustrativo entre las entidades autor y libro, un autor escribe como mínimo un libro y como máximo varios libros (1,n) y, por su parte, un libro puede ser escrito por un autor o por varios (1,n).

La cardinalidad general de la relación autor – escribe – libro es M:N (de muchos a muchos, ya que se cogen los valores máximos de las cardinalidades particulares de autor y libro).

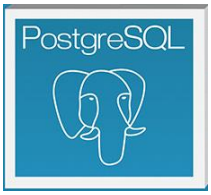
INSTITUCION EDUCATIVA TECNICO "DOMINGO BELISARIO GOMEZ"
Autorizada según Resolución N°. 0480 26 abril de 2004
DANE 119100002474 Nit. 891 501 59-07
Pre-escolar, Primaria, Básica Secundaria y Media Técnica.
Especialidades: Sistemas, Comercio e Industria: Ebanistería y Metalistería
Carrera 3 Nª 11-10 Celular 313 601 05 97 Barrio San Francisco Bolívar – Cauca

AHORA SI A PRACTICAR

1. ¿Después de leer la introduccion de la guia de aprendizaje con sus propias palabras defina que es una base de datos?
2. De un ejemplo de una base de datos diferente a los presentados en la guía.
3. ¿Por qué se crearon las bases de datos?
4. ¿Qué problemas se pueden encontrar en las bases de datos?

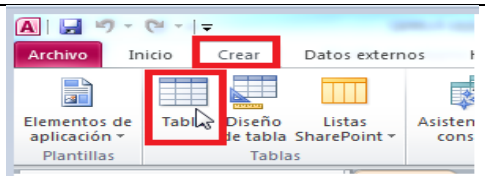
Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje: identificar un sistema de gestión de base de datos (SGBD).

5. Completar la siguiente tabla de los sistemas de gestión de base de datos colocar el nombre y características.

SISTEMAS DE GESTION DE BASE DE DATOS logo	NOMBRE	CARACTERISTICAS
	POSTGRESQL	➢ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios. ➢ Copias de seguridad en caliente (Online/hot backups) ➢ Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.
		
		
		
		

Access		

6. Completar la siguiente tabla.

COMPONENTES DE UNA BASE DE DATOS	DEFINICION	UBICACIÓN EN ACCESS POR DONDE SE CREA (PANTALLAZO) esta columna aplica solo si tienes PC	EJEMPLO
TABLA	Son objetos que se utilizan para almacenar datos referentes a un tema particular.		Clientes, Proveedores, Servicios, Alumnos, etc

7. Completar la siguiente tabla

Tipo de relación entre tablas	Concepto	Ejemplo
Uno a varios:	Los datos de un campo pueden tener múltiples registros en el campo de la otra tabla.	La tabla ventas con empleados por que un empleado puede realizar varias ventas pero las ventas solo se pueden realizar por un empleado ósea de uno a varios

8. Completar la siguiente tabla.

CLAUSULA SQL	FUNCION	EJEMPLO SINTAXIS

SELECT	Muestra una lista de los campos que contienen datos de interés	SELECT field_1 FROM table_1 WHERE criterion_1;

9. Diseñar un mapa mental que involucre al modelo entidad relación sus elementos y relaciones.

10. Diseñar un modelo entidad relación, teniendo en cuenta los pasos para diseñar una Base de Datos y plasmarlo en un video, nota de voz o trabajo escrito, para un base de datos de un colegio y un parqueadero.

LOS TRABAJOS SOBRESALIENTES EL INSTRUCTOR EXCLUIRA A LOS APRENDICES DE PRESENTAR LA EVALUACION SOBRE BASE DE DATOS.