

DEFINICIÓN BASE DE DATOS

- ▶ El concepto de base, procedente del latín *basis*, tiene múltiples usos. El término puede utilizarse con referencia al sostén o fundamento de algo.
- ▶ Un dato se entiende como información concreta sobre hechos, elementos, etc., que permite estudiarlos, analizarlos o conocerlos

DEFINICIÓN BASE DE DATOS

- Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso
- Se llama base de datos, o también *banco de datos*, a un conjunto de información perteneciente a un mismo contexto, ordenada de modo sistemático para su posterior recuperación, análisis y/o transmisión
- Una base de datos es el conjunto de datos informativos organizados en un mismo contexto para su uso y vinculación.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD O DBMS)-DEFINICIÓN

Es un software que administra y gestiona la información que contiene una base de datos. A través de él se maneja todo el acceso a la base de datos con el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones.



SISTEMA DE GESTION DE BASES DE DATOS(SGBD O DBMS)-OBJETIVOS

Objetivos principales de un SGBD:

► Independencia de datos:

- Los programas de aplicación deben verse afectados lo menos posible por cambios efectuados, en datos que no usan.

► Integridad de datos:


- La información que se almacena en la BD debe cumplir ciertos requisitos de calidad, es decir que en el momento de almacenar los datos, estos se deben almacenar debidamente y que posteriormente no se deterioren.

► Seguridad de los datos:

- A la información almacenada en la BD sólo pueden acceder las personas autorizadas y de la forma autorizada.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD O DBMS)-SERVICIOS

Los servicios que debe ofrecer un SGBD son:

- ▶ Creación y definición de la base de datos.
 - ▶ Manipulación de los datos.
 - ▶ Acceso a los datos.
 - ▶ Mantener la integridad y consistencia de los datos.
 - ▶ Mecanismos de copias de respaldo y de recuperación
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-ARQUITECTURA

Cuatro niveles:

1. Nivel interno o físico

- Descripción de la representación en la memoria externa del ordenador de los datos del esquema lógico, sus interrelaciones y los instrumentos para acceder a ellos.

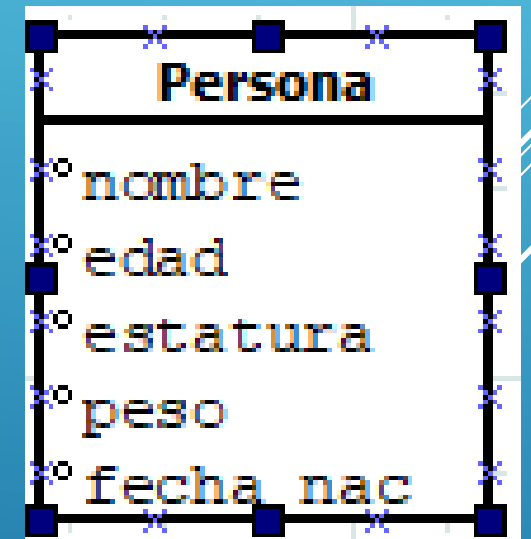
Persona	
° nombre	varchar
° edad	int
° estatura	double
° peso	double
° fecha nac	date

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-ARQUITECTURA

Cuatro niveles:

2. Nivel lógico

- En este esquema lógico se representan las entidades y relaciones de acuerdo a las características de dicho modelo sin entrar todavía en detalles de representación física

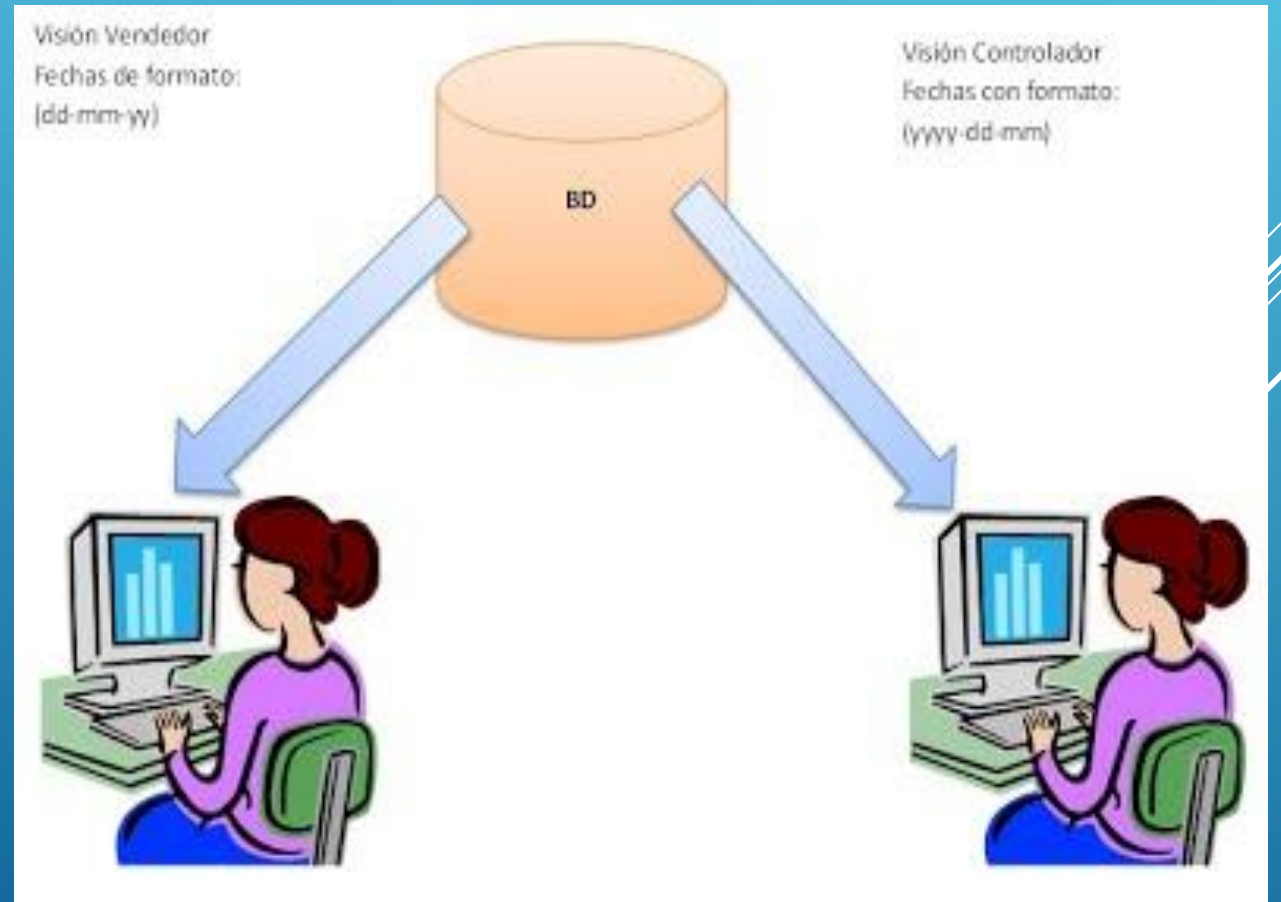


SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-ARQUITECTURA

Cuatro niveles:

3. Nivel externos o de visión

- Cada uno de ellos describe los datos y relaciones entre ellos de interés para una aplicación dada, estos esquemas pueden verse como subconjuntos de Modelo Lógico de la BD.



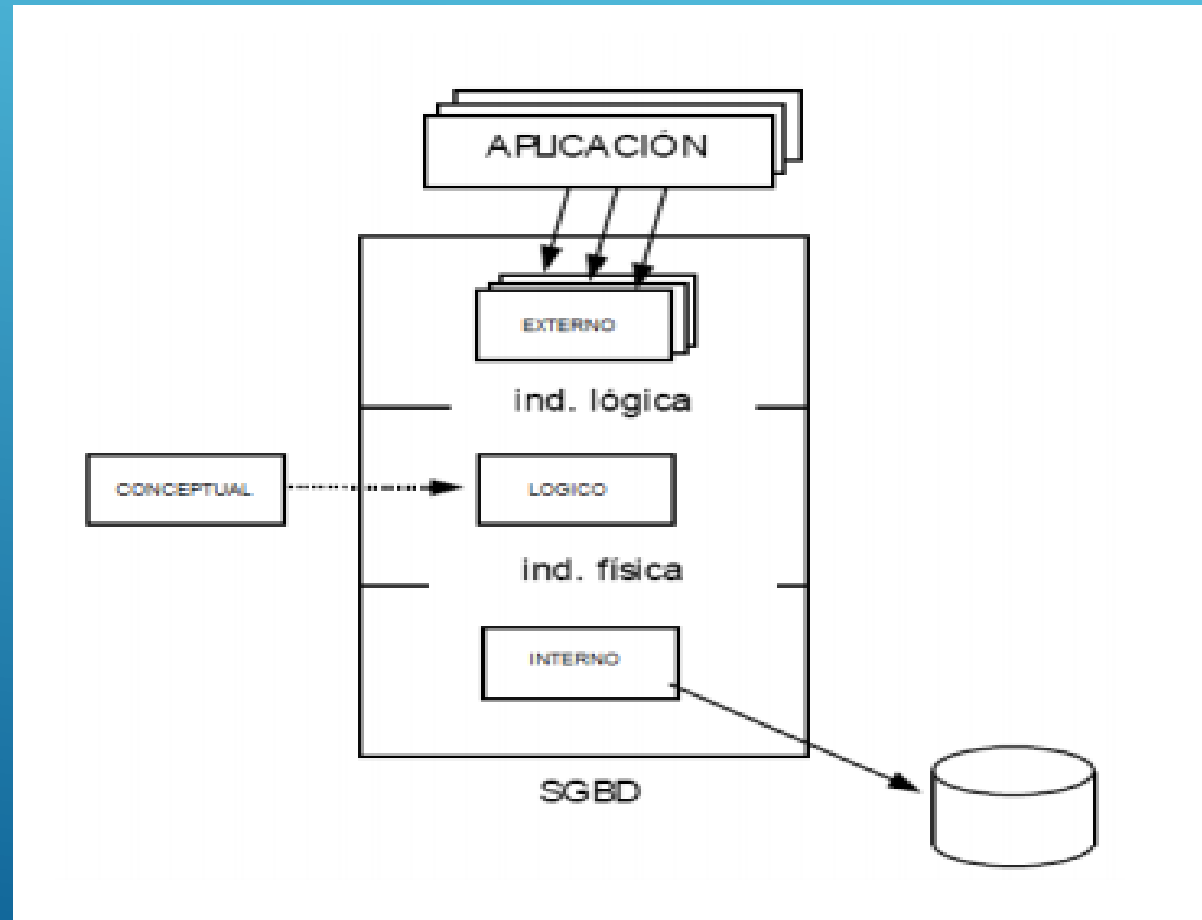
SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-ARQUITECTURA

Cuatro niveles:

4. Nivel conceptual

- ▶ Visión desde un punto de vista organizativo, independiente del SGBD que se utilice, e incluso de la utilización o no de sistemas de bases de datos. En este nivel se describe la información de la organización (objetos y relaciones) desde un punto de vista no informático.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-REPRESENTACIÓN – ARQ-NIV-4



SISTEMA DE GESTION DE BASES DE DATOS(SGBD)- TRANSFORMACIONES

Un SGBD debe poder transformar cualquier petición de usuario(esquema externo) a una petición expresada en termino de esquema conceptual, para finalmente ser una petición en el esquema interno que se procesara sobre la base de datos almacenada.

Se le denomina transformaciones al proceso de transformar peticiones y resultados de un nivel a otro.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)- TRANSFORMACIONES

Un SGBD es capaz de realizar los siguientes pasos:

- El usuario solicita unos datos y crea una consulta.
- El SGBD verifica y acepta el esquema externo para ese usuario.
- Transforma la solicitud al esquema conceptual.
- Verifica y acepta el esquema conceptual.
- Transforma la solicitud al esquema físico o interno.
- Selecciona la o las tablas implicadas en la consulta y ejecuta la consulta.
- Transforma del esquema interno al conceptual, y del conceptual al externo.
- Finalmente, el usuario ve los datos solicitados.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-LENGUAJES

Los lenguajes de los SGBD son herramientas que permiten a todo tipo de usuario realizar ciertas operaciones sobre el SGBD. Los tres principales son :

Lenguaje de manipulación de datos (DML)

permite a los usuarios introducir datos para posteriormente realizar tareas de consultas o modificación de los datos que contienen las Bases de Datos.

- ▶ **SELECT:** esta sentencia se utiliza para realizar consultas sobre los datos.
- ▶ **INSERT:** con esta instrucción podemos insertar los valores en una base de datos.
- ▶ **UPDATE:** sirve para modificar los valores de uno o varios registros.
- ▶ **DELETE:** se utiliza para eliminar las finas de una tabla

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-LENGUAJES

Los lenguajes de los SGBD son herramientas que permiten a todo tipo de usuario realizar ciertas operaciones sobre el SGBD. Los dos principales son :

Lenguaje de definición de datos(DDL)

- ▶ Este lenguaje permite a los administradores de un sistema gestor de base de datos, como definir las estructuras que almacenarán los datos así como los procedimientos o funciones que permitan consultarlos.
- ▶ **CREATE:** se usa para crear una base de datos, tabla, vistas, etc.
- ▶ **ALTER:** se utiliza para modificar la estructura, por ejemplo añadir o borrar columnas de una tabla.
- ▶ **DROP:** con esta sentencia, podemos eliminar los objetos de la estructura, por ejemplo un índice o una secuencia.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)-LENGUAJES

Los lenguajes de los SGBD son herramientas que permiten a todo tipo de usuario realizar ciertas operaciones sobre el SGBD. Los dos principales son :

Lenguaje de Control de Datos (DCL)

- ▶ Esta lenguaje permite al Administrador del sistema gestor de base de datos, controlar el acceso a los objetos, es decir, podemos otorgar o denegar permisos a uno o más roles para realizar determinadas tareas.

SISTEMA DE GESTION DE BASES DE DATOS(SGBD)

Algunos de los SGBD más populares son:

- ▶ MYSQL (SQL)
 - ▶ POSTGRES (SQL)
 - ▶ ORACLE (SQL)
 - ▶ MSSQL (SQL)
 - ▶ SQLITE (SQL)
 - ▶ Apache Cassandra(NOSQL)
 - ▶ MONGODB (NOSQL)
- 
- Several white diagonal lines of varying lengths and thicknesses are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

ATOMICITY,CONSISTENCY, ISOLATION, DURABILITY

Propiedades ACID

- Atomicidad(Atomicity): cualquier cambio de estado que produce una transacción es atómico. Es decir, ocurren todos o no ocurre ninguno. En otras palabras, esta propiedad asegura que una operación se realiza o no se realiza, por lo tanto no puede quedar el sistema a medias.
- Consistencia(Consistency): propiedad que asegura que una transacción no romperá con la integridad de una base de datos, pues respeta todas las reglas y directrices de ésta.
- Aislamiento(Isolation): propiedad que asegura que no se afectarán entre sí las transacciones. En otras palabras, dos o más transacciones sobre los mismos datos no generarán un problema.
- Durabilidad(Durability): propiedad que asegura la persistencia de una transacción, es decir, una vez que la transacción quedó aceptada no podrá deshacerse aunque falle el sistema.

TIPOS DE BASES DE DATOS

► Bases de datos estáticas

- Bases de datos de sólo lectura, utilizadas primordialmente para almacenar Datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

► Bases de datos dinámicas

- Bases de datos donde la Información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y adición de datos, además de las operaciones fundamentales de consulta.

TIPOS DE BASES DE DATOS

► Bases de datos bibliográficas

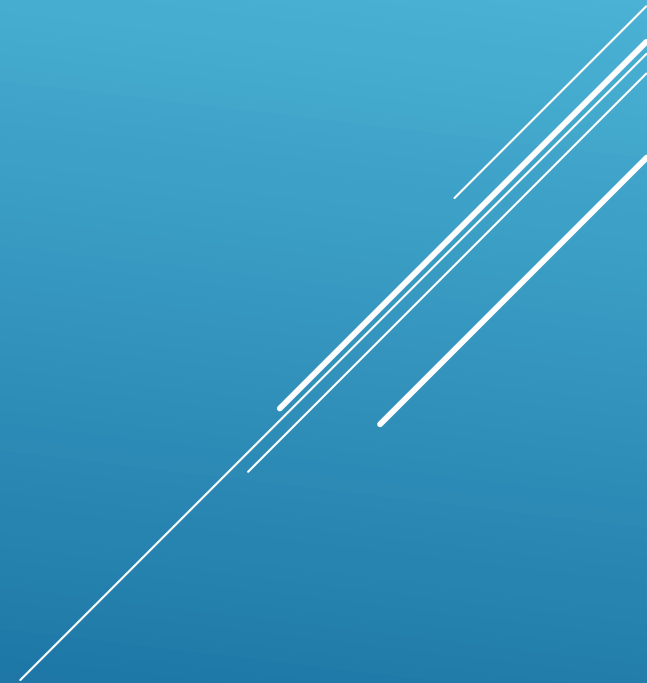
- Solo contienen un representante de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque si no, estaríamos en presencia de una base de datos a texto completo

► Bases de datos documentales

- Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

MODELOS DE BASES DE DATOS

Podemos entender un modelo de bases de datos como una forma de organizar y ordenar la información en un BD para que esta sea accesible para nosotros. Debido a que no existe un sistema de BD perfecto se han creado varios modelos para elegir la estructura que más se adapte a nuestras necesidades.



MODELOS DE BASES DE DATOS – LOS MAS COMUNES

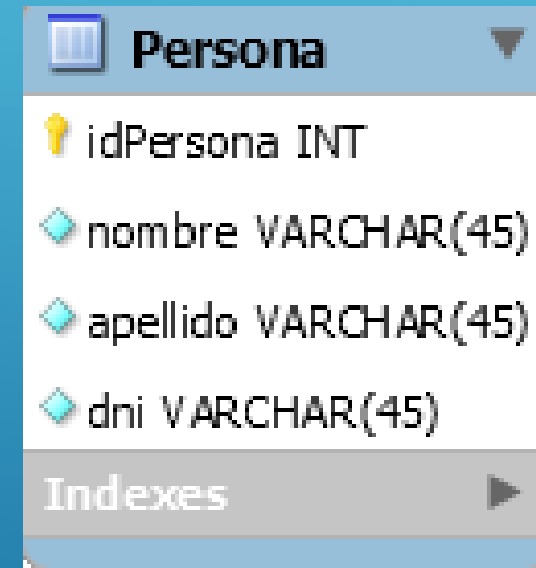
- ▶ **Las bases de datos jerárquicas** construyen una estructura de jerarquía con los datos que permite una estructuración muy estable cuando gestionamos una gran cantidad de datos muy interrelacionados.
- ▶ **Las bases de datos en red** derivan de las jerárquicas pero mejoran la gestión de datos redundantes manteniendo su rendimiento en consultas de datos.
- ▶ **Las bases de datos transaccionales** están diseñadas para el envío y recepción de datos a grandes velocidades y de forma continua. Su único fin es la recepción y envío de información pero la gestión de almacenamiento o redundancia están fuera de su propósito.

MODELOS DE BASES DE DATOS – LOS MAS COMUNES

- ▶ **Las bases de datos orientadas a objetos** han surgido como concepto tras la aparición de los sistemas de programación orientada a objetos.
- ▶ **Las bases de datos relacionales** Este es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia.
- ▶ **Las bases de datos documentales** están especializadas en el almacenamiento de textos completos, por lo que facilitan el tratamiento informatizado de grandes cadenas de caracteres.

FUNDAMENTOS BASE DE DATOS RELACIONAL

- ▶ **ENTIDAD:** representa un objeto, cosa o concepto del mundo real (abstracción):
 - ▶ Carro
 - ▶ Persona
 - ▶ Compra

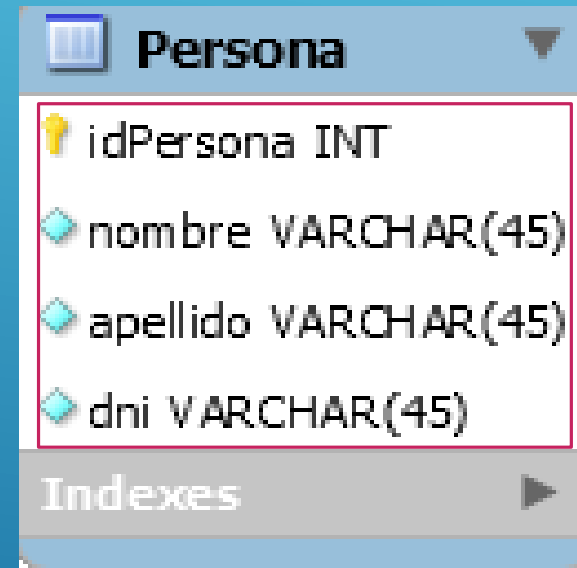


A screenshot of a database management tool showing the definition of a table named 'Persona'. The table has four columns: 'idPersona' (INT, primary key), 'nombre' (VARCHAR(45)), 'apellido' (VARCHAR(45)), and 'dni' (VARCHAR(45)). Below the columns is a section for 'Indexes'.

Persona	
idPersona	INT
nombre	VARCHAR(45)
apellido	VARCHAR(45)
dni	VARCHAR(45)
Indexes	

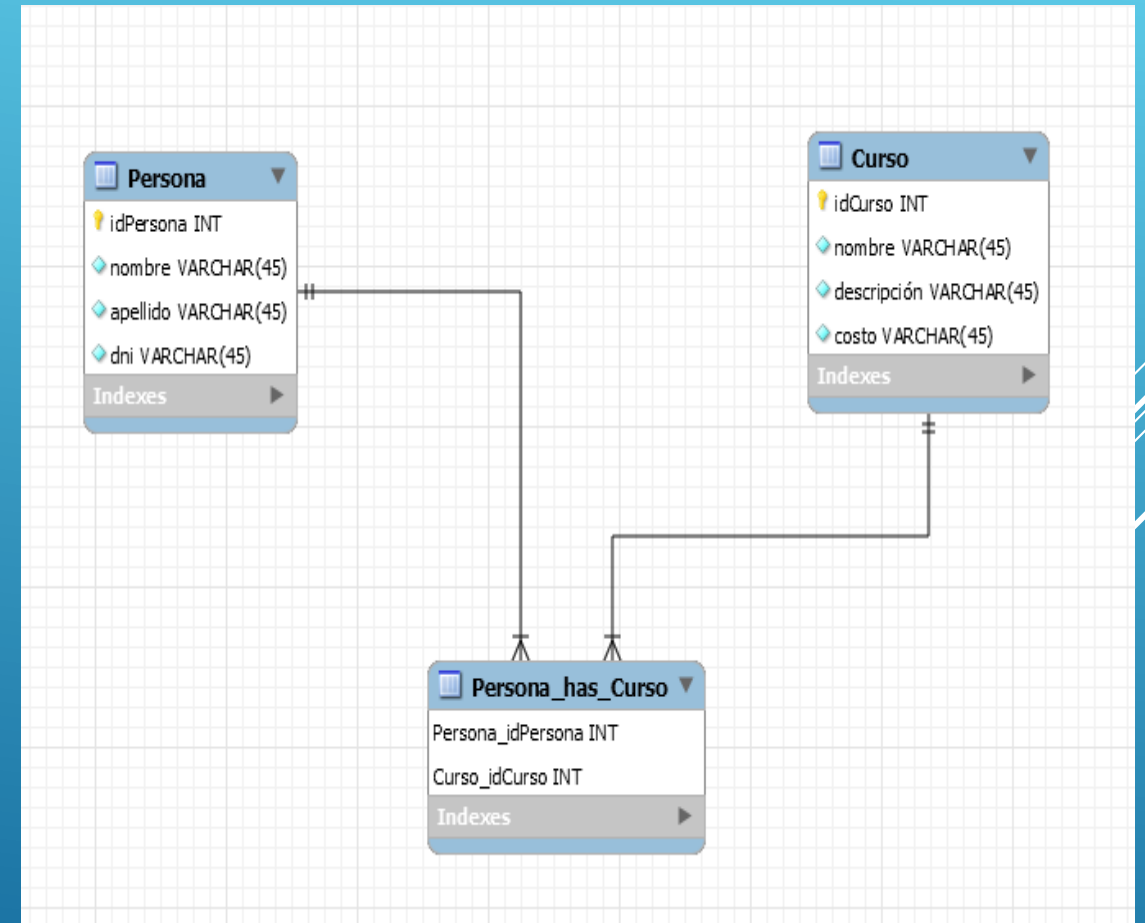
FUNDAMENTOS BASE DE DATOS RELACIONAL

- ▶ **ATRIBUTOS:** son las características que definen o identifican una entidad.
 - ▶ Nombre
 - ▶ Peso
 - ▶ Estatura
 - ▶ Fecha



FUNDAMENTOS BASE DE DATOS RELACIONAL

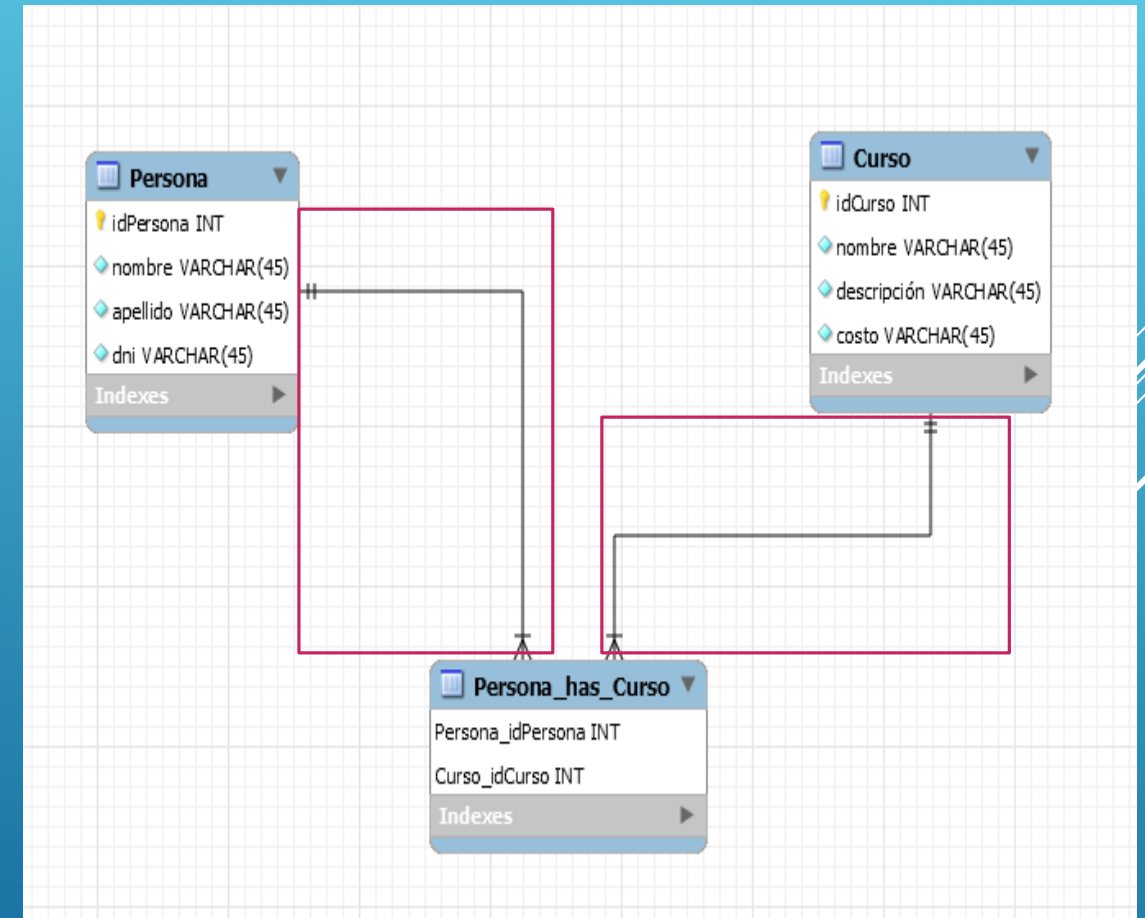
- **RELACIÓN:** describe cierta dependencia entre entidades y también permita la asociación entre estas.



FUNDAMENTOS BASE DE DATOS

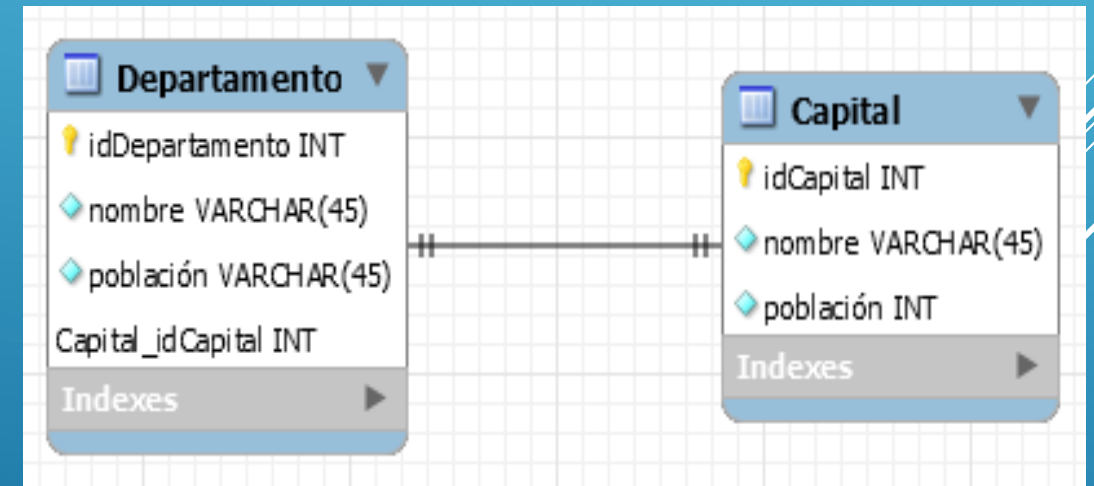
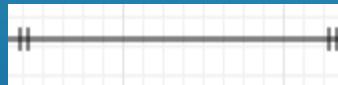
RELACIONAL- CARDINALIDAD

- **CARDINALIDAD:** indica el número de entidades con que puede estar relacionado una entidad dada.



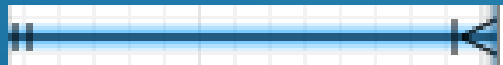
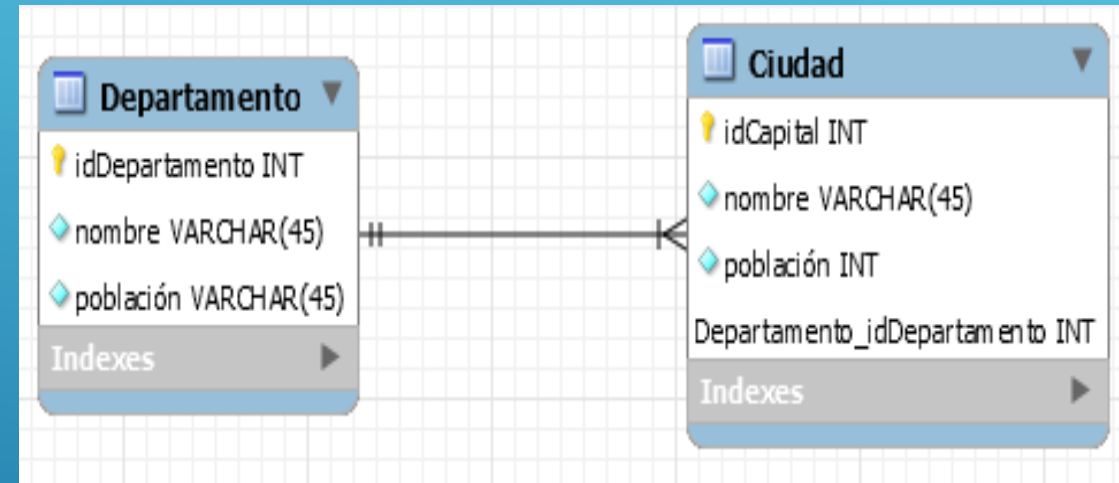
FUNDAMENTOS BASE DE DATOS RELACIONAL- CARDINALIDAD (UNO A UNO)

- **UNO A UNO (1:1):** un registro de una entidad A solo se relaciona con un registro de la entidad B.



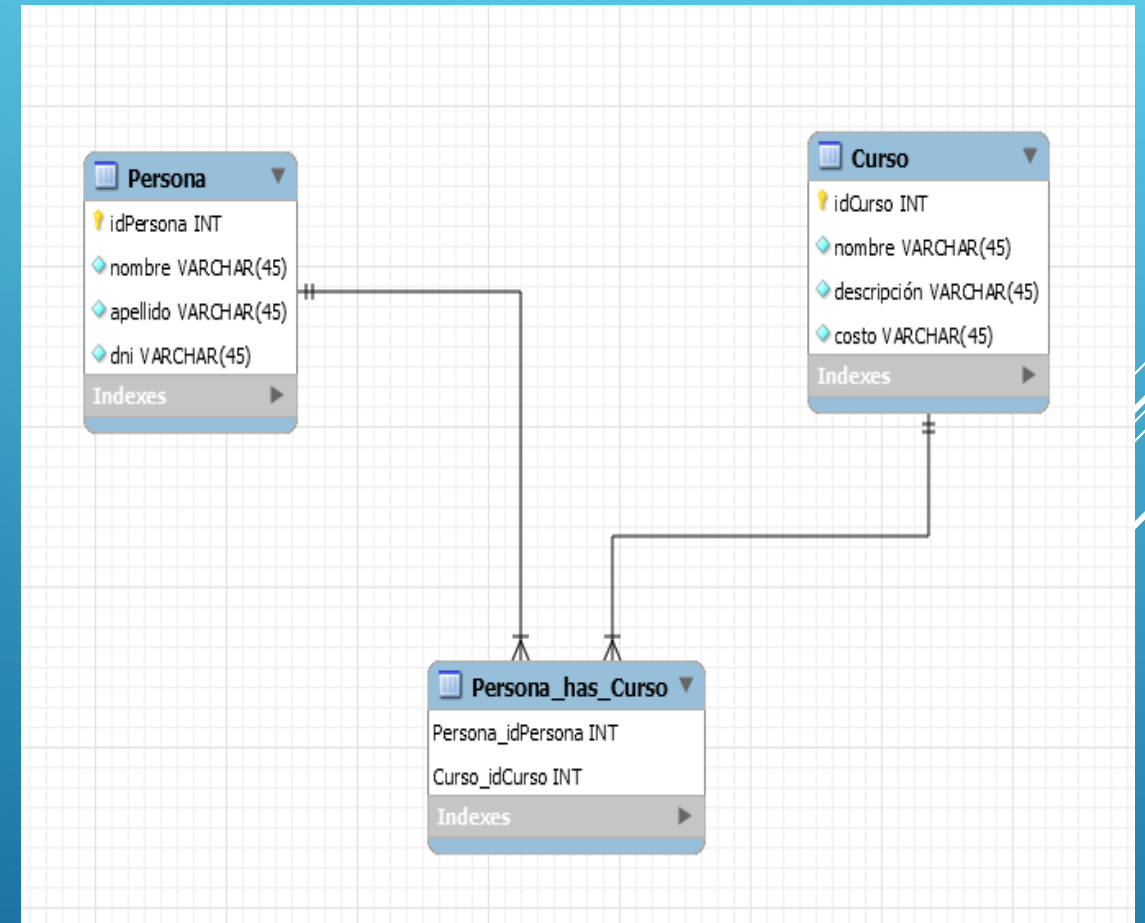
FUNDAMENTOS BASE DE DATOS RELACIONAL- CARDINALIDAD (UNO A MUCHOS)

- **UNO A MUCHOS (1:N):** Un registro en una entidad en A se relaciona con cero o muchos registros en una entidad B. Pero los registros de B solamente se relacionan con un registro en A.



FUNDAMENTOS BASE DE DATOS RELACIONAL - CARDINALIDAD (MUCHOS A MUCHOS)

- **MUCHOS A MUCHOS:** una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa



FUNDAMENTOS BASE DE DATOS RELACIONAL - CLAVES

- ▶ **Clave Primaria:** Es un identificador ÚNICO para cada registro de una entidad.
- ▶ **Clave foránea:** se puede entender como un descriptor, dado que no es necesariamente única. La clave foránea es una clave primaria de una entidad usada en otra entidad.

DIAGRAMA ENTIDAD-RELACIÓN (COMPONENTES)

Entidad: abstracción de un objeto, persona o concepto .

Conjunto de entidades: varias entidades.

Entidad fuerte: se puede definir únicamente con sus atributos.

Entidad débil: depende de una o varias entidades.

Entidad asociativa: se usa como tabla pivote cuando hay una relación de N:M

Relación: indica el cómo las relaciones actúan entre sí.

Atributo: propiedad o característica de una entidad

Atributos múltiples: varios valores en un atributo, por ejemplo varios números de telefonos

Atributo compuesto: sub-atributos que surgen de un atributo.

Atributo derivados: surge de otros atributos (edad a través de fecha)

DIAGRAMA ENTIDAD-RELACIÓN “CHEN” (COMPONENTES)

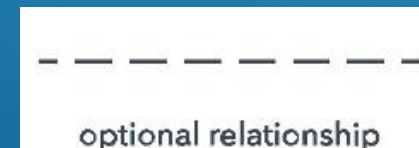
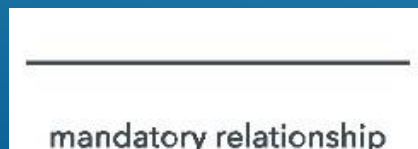
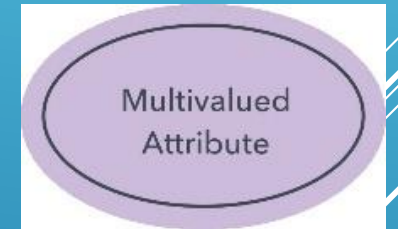
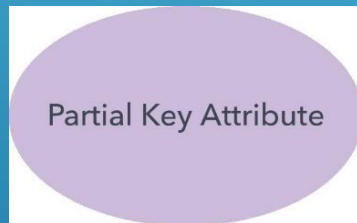
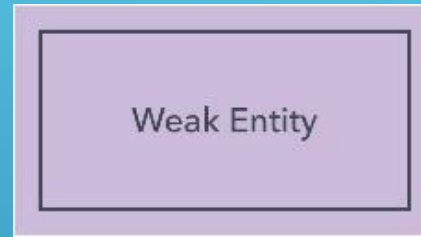


DIAGRAMA ENTIDAD-RELACIÓN “CHEN”- EJEMPLO

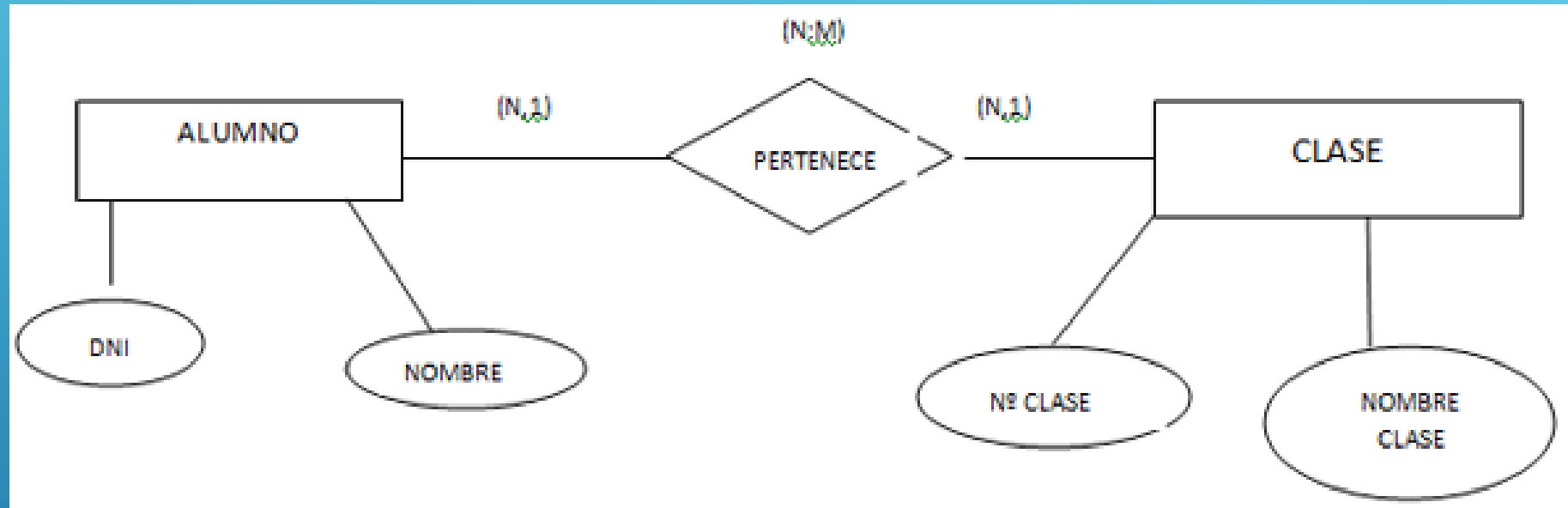


DIAGRAMA ENTIDAD-RELACIÓN “BARKER” (COMPONENTES)

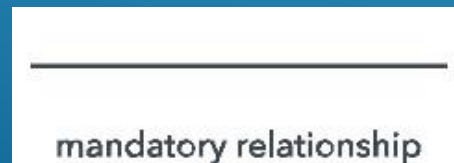
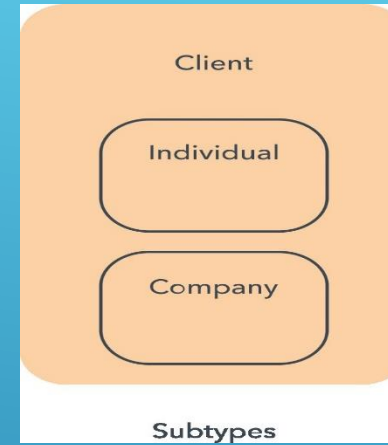
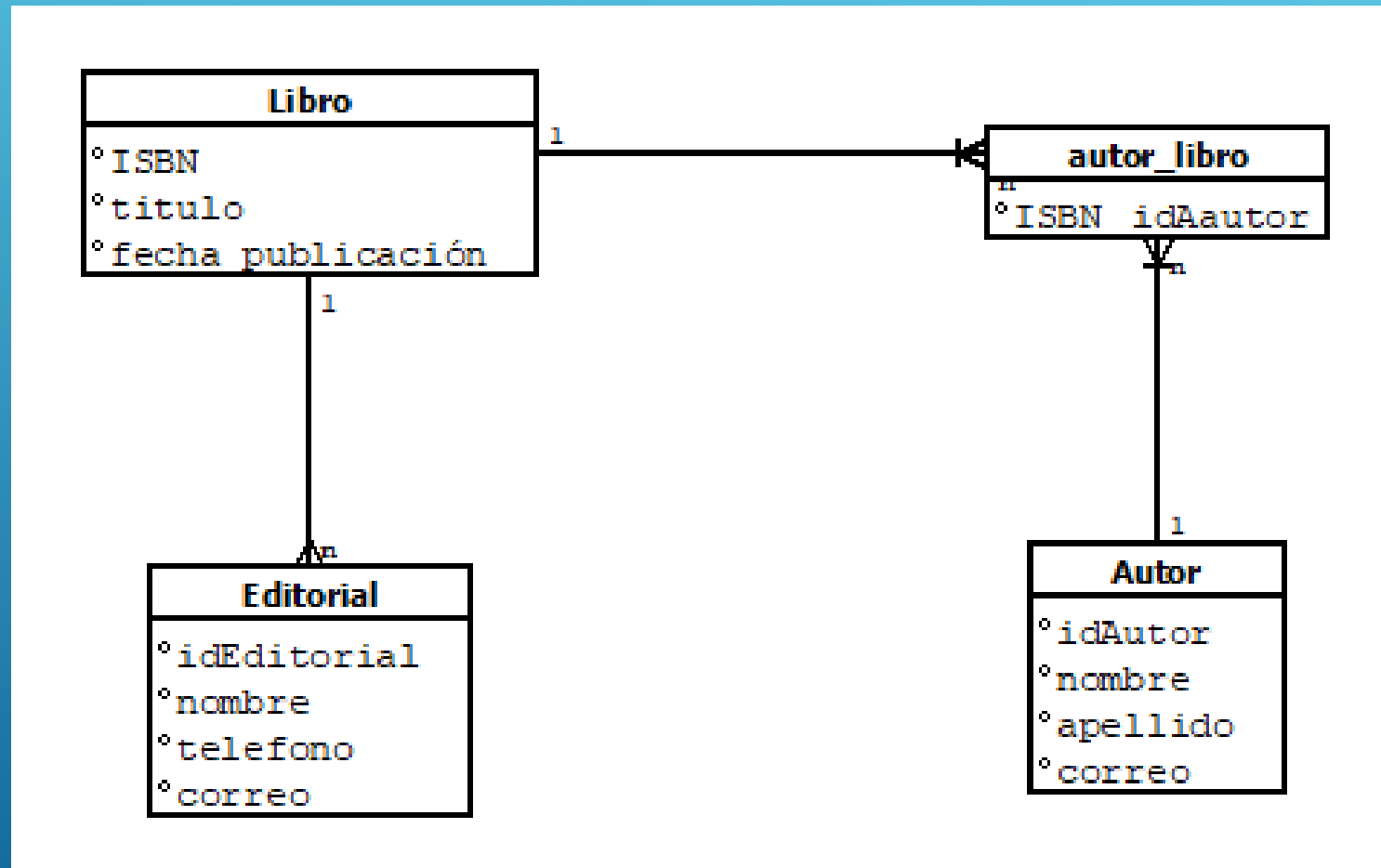


DIAGRAMA ENTIDAD-RELACIÓN “BARKER” - EJEMPLO



SQL Y NOSQL

SQL(Lenguaje de consulta estructurado): es un tipo de lenguaje vinculado con la gestión de BD relacionales que permite la especificación de distintas clases de operaciones entre estas. Gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla.

NOSQL(Lenguaje de consulta no estructurado): es un tipo de lenguaje vinculado con la gestión de BD que a diferencia del SQL no necesita una estructura de datos fijos como las tablas, debido a esto no garantiza completamente las características ACID. El NOSQL suele ser usado para trabajar con grandes cantidades de datos a diferencia del SQL que no está diseñado para esto. EL NOSQL se suele usar cuando se necesita el manejo de grandes cantidades de datos(Big Data), que un sistema tenga escalabilidad horizontal(Software Hardware distribuido) entre otros, pero esto no es un requisito para poder usarlo.

NORMALIZACIÓN DE BASES DE DATOS

- La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

NORMALIZACIÓN DE BASES DE DATOS

– PRIMERA FORMA NORMAL (1FN)

- ▶ Elimine los grupos repetidos de las tablas individuales.
- ▶ Cree una tabla independiente para cada conjunto de datos relacionados.
- ▶ Identifique cada conjunto de datos relacionados con una clave principal.

No usar varios campos en una sola tabla para almacenar datos similares.

NORMALIZACIÓN DE BASES DE DATOS

– SEGUNDA FORMA NORMAL (2FN)

- ▶ Cree tablas independientes para conjuntos de valores que se apliquen a varios registros.
- ▶ Relacione estas tablas con una clave externa.

Los registros no deben depender de nada que no sea una clave principal de una tabla, una clave compuesta si es necesario.

NORMALIZACIÓN DE BASES DE DATOS

– TERCERA FORMA NORMAL (3FN)

- ▶ Elimine los campos que no dependan de la clave.

Los valores de un registro que no sean parte de la clave de ese registro no pertenecen a la tabla. En general, siempre que el contenido de un grupo de campos pueda aplicarse a más de un único registro de la tabla, considere colocar estos campos en una tabla independiente.

NORMALIZACIÓN DE BASES DE DATOS EJEMPLO

Estos pasos demuestran el proceso de normalización de una tabla de alumnos ficticia.

1.Tabla sin normalizar

Nombre_al	Tutor	Despacho-Tut	Clase1	Clase2	Clase3
Juan	García	412	101-07	143-01	159-02
camila	Díaz	216	201-01	211-02	214-01

NORMALIZACIÓN DE BASES DE DATOS – PRIMERA FORMA NORMAL (1FN) - EJEMPLO

2.1FN: no hay grupos repetidos Puesto que un alumno tiene varias clases, estas clases deben aparecer en una tabla independiente. Los campos Clase1, Clase2 y Clase3 de los registros anteriores son indicativos de un problema de diseño.

Alumno

Nº alumno	Nombre_al	Tutor	Despacho-Tut	Nº clase
1022	Juan	García	412	101-07
1022	Juan	García	412	143-01
1022	Juan	García	412	159-02
4123	Camila	Díaz	216	201-01
4123	Camila	Díaz	216	211-02
4123	Camila	Díaz	216	214-01

NORMALIZACIÓN DE BASES DE DATOS – SEGUNDA FORMA NORMAL (2FN) - EJEMPLO

3. 2FN: eliminar los datos redundantes

Observe los diversos valores de N° clase para cada valor de N° alumno en la tabla anterior. N° clase no depende funcionalmente de N° alumno (la clave principal), de modo que la relación no cumple la segunda forma normal.

Las dos tablas siguientes demuestran la segunda forma normal:

Alumnos

N° alumno	Nombre_al Tutor		Despacho -Tut
1022	Juan	García	412
4123	Camila	Díaz	216

Registro

N° alumno	N° clase
1022	101-07
1022	143-01
1022	159-02
4123	201-01
4123	211-02
4123	214-01

NORMALIZACIÓN DE BASES DE DATOS – TERCERA FORMA NORMAL (3FN) - EJEMPLO

4. 3FN: eliminar los datos no dependientes de la clave

En el último ejemplo, Despacho-Tut (el número de despacho del tutor) es funcionalmente dependiente del atributo Tutor. La solución es pasar ese atributo de la tabla Alumnos a la tabla Personal, según se muestra a continuación:

Alumnos

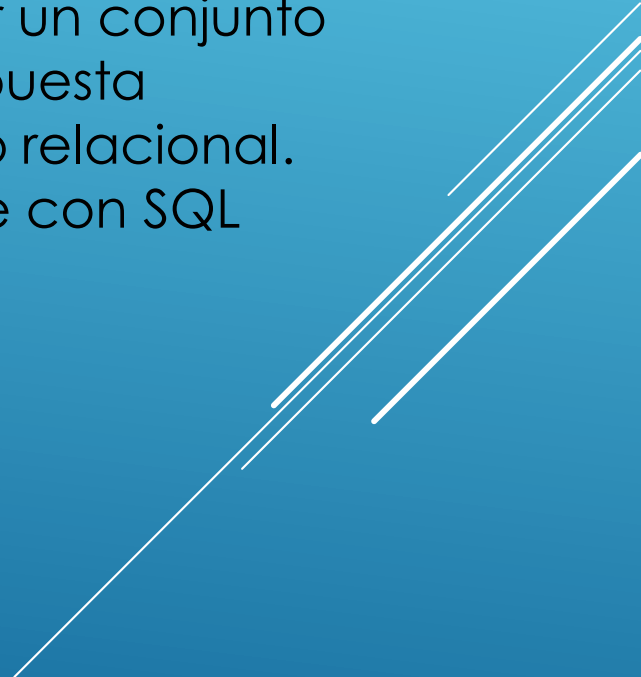
Nº alumno	Nombre_al	NºTutor
1022	Juan	1
4123	Camila	2

Personal


Nºtutor	Nombre	Despacho
1	García	412
2	Díaz	216

INTRODUCCIÓN AL ÁLGEBRA RELACIONAL

El álgebra relacional es un lenguaje de consulta procedural, es decir un conjunto de operaciones que describen paso a paso cómo computar una respuesta sobre las relaciones tal y como estas han sido definidas en el modelo relacional. El álgebra relacional es la base del SQL, pero no todo lo que se hace con SQL se puede hacer con el álgebra relacional.

Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

INTRODUCCIÓN AL ÁLGEBRA RELACIONAL


- Comandos para inserción, borrado o modificación de datos.
 - Capacidades aritméticas
 - Asignación y comandos de impresión: es posible imprimir una relación construida por una consulta y asignar una relación calculada a un nombre de relación.
 - Funciones agregadas: Operaciones tales como promedio (average), suma (sum), máximo (max), etc. se pueden aplicar a las columnas de una relación para obtener una cantidad única.
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

INTRODUCCIÓN AL ÁLGEBRA RELACIONAL (OPERADORES)

Operaciones básicas:

- Selección
- Proyección
- Producto
- Unión
- Diferencia

Operaciones No básicas o derivadas:

- Intersección
 - Unión natural
 - División
 - Agrupación
- 
- Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

INTRODUCCIÓN AL ÁLGEBRA RELACIONAL (OPERADORES)

Selección(σ): el operador de selección opta por tuplas que satisfagan cierto predicado

Proyección(Π): la operación de proyección permite seleccionar algunos atributos de una relación

Producto(X): es una operación que toma dos relaciones y obtiene una nueva relación cuyas tuplas son la concatenación de las tuplas de la primera relación con las tuplas de la segunda.

UNIÓN(U): la unión de dos relaciones devuelve una nueva relación que contiene todas las tuplas que aparecían en cualquiera de las relaciones originales, o en ambas.

Diferencia(-): produce el conjunto de todas las tuplas T que pertenecen a A y no pertenecen a B.

NOTA: En base de datos relacional, un registro (también llamado fila o **tupla**) representa un objeto único de datos implícitamente estructurados en una **tabla**.

MONGODB

- **MongoDB** es una base de datos NoSQL escrita en c++ y orientado a documentos, debido a esto mongo es muy popular a la hora de trabajar con grandes cantidades de datos (bigdata).
- **Colección:** La colección es un grupo de documentos de MongoDB. Es el equivalente de una tabla RDBMS. Existe una colección dentro de una sola base de datos. Las colecciones no imponen un esquema. Los documentos dentro de una colección pueden tener diferentes campos. Normalmente, todos los documentos de una colección tienen un propósito similar o relacionado.
- **Documento:** es un conjunto de pares clave-valor. Los documentos tienen esquema dinámico. El esquema dinámico significa que los documentos en la misma colección no necesitan tener el mismo conjunto de campos o estructura, y los campos comunes en los documentos de una colección pueden contener diferentes tipos de datos.(Formato JSON)

MONGODB

STRING: este es el tipo de datos más utilizado para almacenar los datos. La cadena en MongoDB debe ser UTF-8 válida.

Integer: este tipo se utiliza para almacenar un valor numérico. El número entero puede ser de 32 o 64 bits, dependiendo de su servidor.

Boolean: este tipo se utiliza para almacenar un valor booleano (verdadero / falso).

Object ID: este tipo de datos se utiliza para almacenar la ID del documento.

Binary Data: este tipo de datos se utiliza para almacenar datos binarios.

Code: este tipo de datos se utiliza para almacenar el código JavaScript en el documento.

MONGODB

Double: este tipo se utiliza para almacenar valores de punto flotante.

Mín. / Máx key: Este tipo se utiliza para comparar un valor con los elementos BSON más bajos y más altos.

Arrays: este tipo se usa para almacenar arrays o listas o múltiples valores en una clave.

Regular expression: este tipo de datos se utiliza para almacenar expresiones regulares.

Timestamp - ctimestamp: Esto puede ser útil para grabar cuando un documento ha sido modificado o agregado.

Object: este tipo de datos se utiliza para documentos incrustados.

Null: este tipo se utiliza para almacenar un valor nulo.

MONGODB

Symbol: este tipo de datos se utiliza de forma idéntica a una cadena; sin embargo, generalmente se reserva para idiomas que usan un tipo de símbolo específico.

Fecha: este tipo de datos se utiliza para almacenar la fecha u hora actual en formato de hora UNIX. Puede especificar su propia fecha y hora al crear el objeto de Fecha y pasar día, mes y año.

BIBLIOGRAFÍA

- ▶ https://www.ecured.cu/Bases_de_datos
- ▶ <https://todopostgresql.com/diferencias-entre-ddl-dml-y-dcl/>
- ▶ <http://www.alegsa.com.ar/Dic/acid.php>
- ▶ <https://support.microsoft.com/es-co/help/283878/description-of-the-database-normalization-basics>
- ▶ <https://definicion.de/sql/>
- ▶ https://www.tutorials.com/mongodb/mongodb_datatype.htm