

INTRODUCCIÓN A LAS BASES DE DATOS

El curso de Introducción a las Base de datos ha sido un viaje a través de los fundamentos de estos repositorios de datos. Comenzaste por identificar los datos en un contexto relevante para la solución de tu problema y su posterior agrupación en entidades y sus atributos y tipos.

Después de conocer la arquitectura de un sistema de base de datos, revisamos los paradigmas más comunes, agrupados en Relacionales y No Relacionales, también conocidos como SQL y NoSQL, respectivamente.

No sólo pudiste distinguir entre estos tipos de base de datos sino que también manipulaste las mismas usando herramientas y estructuras de manejadores, tales como MySQL, para bases de datos relacionales, y MongoDB, para las no relacionales. Para finalizar, exploramos las formas de documentar tus datos usando diagramas entidad/relación para los datos SQL y las estructuras jerárquicas basadas en árboles de documentos para los datos no relacionales tipo JSON.

Luego de haber transitado por este camino introductorio de la bases de datos has desarrollado la capacidad de:

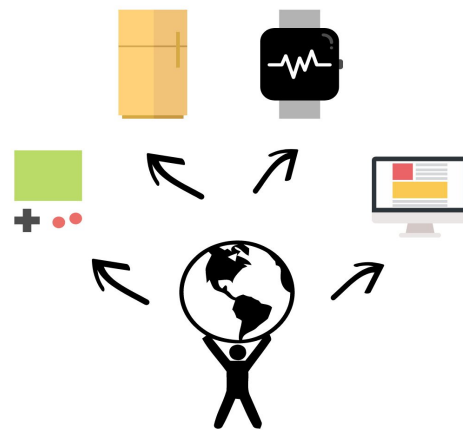
- Interpretar los tipos de bases de datos para establecer las relaciones entre entidades de un sistema.
- Crear la estructura de base de datos que se adapta a la solución basado en mejores prácticas
- Formular diagramas de bases de datos usando notaciones estándares del mercado
- Diferenciar los sistema manejadores de base de datos con la finalidad de familiarizarse con los productos existentes en el mercado.

Ahora estás listo para continuar tu travesía hacia conceptos más avanzados de las bases de datos en los próximos cursos. Pero primero, te dejamos un resumen de lo que hemos cubierto en la Introducción a las Bases de Datos.



UNIDAD 1: INTRODUCCIÓN A LOS DATOS

Nos encontramos inmersos en un mundo donde la tecnología se encuentra donde quiera en nuestro entorno, desde un teléfono móvil hasta un refrigerador inteligente, y todos utilizan algo en común: **datos e información**. Existe además una necesidad creciente de almacenar dicha información para su posterior consulta y utilización.



Las bases de datos, las cuales **son repositorios de datos donde se almacena la información**, juegan un rol importante en la preservación de los datos.

Un paso importante en el proceso de diseño e implementación de una base de datos, es poder saber cómo discriminar entre aquellos datos que son relevantes o no en un contexto dado, en la solución de un problema o la representación de una situación determinada. Es por esto que debemos conocer el funcionamiento de las bases de datos, qué herramientas son necesarias para su manejo y cómo interactúan los sistemas y las bases de datos existentes.

Un sistema de base de datos consta de un **software (aplicación)** desarrollado en cualquier lenguaje de programación, por ejemplo, Python o Java. Por otro lado, tenemos el **sistema manejador de base de datos**, tal como MySQL, encargado de la gestión de la misma.

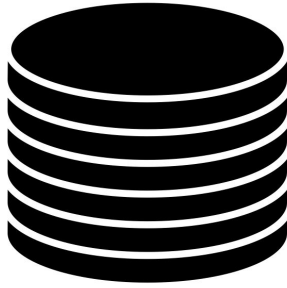
Internamente tenemos un **conector o driver**, el cual es un programa que sirve de puente entre el software y el manejador de base de datos a través de unos datos de acceso, por ejemplo, el driver Python-MySQL, que facilita la conexión entre una aplicación desarrollada en Python y una base de datos MySQL. La conexión se realiza, generalmente, a través de la red **Internet**, funcionando el sistema manejador de base de datos como un servidor. Sin embargo, en algunos casos, la base de datos se puede encontrar de forma local en nuestro computador.

Los datos en una base de datos se organizan de una forma determinada. Antes de adentrarse en estos detalles, debemos conocer cómo vamos identificando y agrupando los datos en estructuras que posteriormente se constituirán en los elementos de la implementación. La definición más básica de una agrupación de los datos es la **entidad**. Cada dato de la entidad se define como un **atributo**, pudiendo una entidad tener muchos atributos. Los atributos, a su vez, tienen tipos, tales como, el tipo **entero** y el tipo **string**.

UNIDAD 2: TIPO DE BASES DE DATOS

La manera de organizar los datos en las bases de datos se agrupan en dos paradigmas conocidos como **Relacional** y **No Relacional**; alguna veces referidos como **SQL** y **NoSQL** respectivamente. En el esquema Relacional los datos se acomodan en estructuras bien definidas denominadas **tablas**, donde se almacenan los datos asociados a una entidad de acuerdo a sus atributos y

correspondientes tipos de datos. Entre los sistemas manejadores de base de datos relacionales tenemos: **MySQL, PostgreSQL, Oracle y Microsoft Access.**



Por su parte, las bases de datos No Relacionales se caracterizan porque las estructuras de los datos no son definidos a priori, con lo cual tenemos la libertad de definir cómo van a estar almacenados. En esta categoría encontramos todas aquellas formas de almacenar datos que no son estructuradas, tal como: **por documentos, por clave/valor, por múltiples columnas y por grafos.** Los sistemas manejadores de base de datos no relacionales más conocidos son: **MongoDB, CouchDB, Memcached, Redis, Cassandra y Neo4j.**

Las **tablas** en las bases de datos relacionales tienen columnas que representan los **campos** de dichas tablas, mientras que las filas representan los **registros**. Una base de datos puede contener más de una tabla, por lo cual se definen los siguientes tipos de **relaciones**; **N a M** (muchos a muchos), **1 a N** (de uno a muchos), **N a 1** (de muchos a uno) y **1 a 1** (de uno a uno). El **lenguaje SQL** nos permite, a través del uso de comandos y sentencias bien definidas, trabajar con estas tablas y sus componentes, es decir, crear, borrar, actualizar e insertar una nueva tabla.

Las bases de datos **orientadas a documentos JSON** constituyen uno de los paradigmas no relacionales comúnmente utilizado. Como lo indica su nombre, los datos se organizan en **documentos**, siendo una agrupación de documentos una **colección**. A diferencia del paradigma relacional, las asociaciones entre documentos se van formando siguiendo una estructura

jerárquica, así un documento puede estar **embebido en otro documento** o un documento puede ser **referenciado por otro**.

UNIDAD 3: SISTEMAS MANEJADORES DE BASES DE DATOS

La consolidación de los datos son las bases de datos, por su parte es el **Sistema Manejador de la Base de Datos (SMBD)** el que nos permite interactuar con las mismas. Este sistema está constituido por el conjunto de herramientas de software que vienen integradas en un producto, con las cuales interactuamos con dicho repositorio. Los sistemas manejadores tienen diversas funcionalidades que delinean la importancia de sus rol en el sistema. Ellos permiten la **creación, actualización, inserción y eliminación** de bases de datos. Los manejadores son también los encargados del **almacenamiento de los datos en estructuras físicas** soportadas por el computador, mientras que permiten su **respaldo** y posterior **recuperación**.

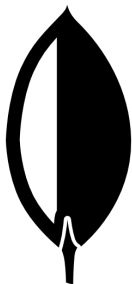
Los manejadores también deben permitir la **configuración de los componentes del sistema** para que pueden conectarse en la red. Los datos en las base de datos deben estar protegidos, el manejador permite **definir roles** y **delimitar el acceso** de acuerdo a los perfiles de los usuarios. Estos sistemas, adicionalmente, pueden soportar la **replicación de las bases de datos** y su sincronización a fin de garantizar la disponibilidad de los datos.



La posibilidad de **monitorear** el uso de la base de datos y los acceso a las mismas también es una funcionalidad de estos sistema. Mientras se accede la base de datos, errores pueden ocurrir, los manejadores deben permitir poder **retroceder una transacción** que ya se ha realizado. La **consistencia** en los

datos es imprescindible; cuando se actualiza la información. El sistema manejador de base de datos asegura que se mantengan la consistencia de acuerdo al conjunto de reglas especificadas.

MySQL es uno de los sistemas manejadores comúnmente utilizado para bases de datos relacionales. Oficialmente MySQL proporciona dos formas principales de interactuar con nuestra base de datos a través de un **cliente MySQL**, que incluye una interfaz para ingresar comandos específicos para manipular las tablas en el repositorio de datos. Alternativamente, se provee de un producto que viene una interfaz de usuario gráfica, que permite realizar dicha manipulación de la base de datos usando una aplicación gráfica denominada **MySQL Workbench**.



Uno de los productos, que constituyen la contraparte de los productos para bases de datos relacionales como MySQL, es **MongoDB**, que es una herramienta para manipular bases de datos no relacionales orientadas a documentos JSON. Similarmente a MySQL, este producto ofrece un **cliente** para interactuar con los documentos y colecciones del repositorio vía comandos, y una interfaz gráfica de usuario, que soporta estas funcionalidades a través de un ambiente gráfico.

Este componente gráfico del producto se llama **MongoDB Compass**.

UNIDAD 4: DIAGRAMAS EN BASE DE DATOS

Tan importante como es lograr una buena implementación de la base de datos también lo es el poder documentar la misma. Existen diversas formas de **documentar nuestros datos**, desde el uso de narrativa hasta el uso de diagramas y lenguajes con notaciones formales bien establecidas. Un **modelo** se define en este contexto como la forma de la estructura de cómo se almacenan los datos, tal como tablas y grafos. Por otra parte, los **diagramas** son una **herramienta comunicacional** para presentar la estructura de los datos sin preocuparnos por los detalles su implementación. Estos son una herramienta muy poderosa para lograr transmitir nuestra ideas acerca de los datos a otros miembros del equipo, a la vez que constituye un punto de colaboración entre las personas trabajando en la solución de un problema en lo que concierne a los datos.

Los diagramas también permiten un entendimiento de la estructura de sistemas existentes, paso necesario cuando se desea emprender una mejora de los mismos. Notemos, sin embargo, que los diagramas pueden ser desventajosas si no estamos familiarizados con la notación que usan y no tenemos la disciplina de actualizarlos regularmente.

Creado a mediados de los años 70s, los **Diagramas de Entidad/Relación** son una de la formas más tradicional y útil de representar datos que se van a estructurar siguiendo el paradigma relacional. Como

indicado en su nombre, este tipo de diagrama se fundamenta en la identificación de la entidades y las relaciones entre ellas.

Los Diagramas de Entidad/Relación usan una **notación** basada en el uso de **formas geométricas básicas**, las cuales se asocian a los elementos de una base de datos, tal como las entidades, atributos y claves primarias. Los diagramas se pueden generar usando papel y lápiz o alguna herramienta de software para dibujar; sin embargo **MySQL Workbench** también proporciona una funcionalidad para dibujar diagramas E/R, con la facilidad adicional de poder generar la base de datos a partir del diagrama y viceversa.

Otro lenguaje estándar que se puede utilizar para el modelado de datos relacionales es el **lenguaje UML** o **lenguaje de modelado unificado**, muy utilizado en lenguajes de programación orientados a objetos. UML sigue una notación estándar bien conocida y amplia, pero para el modelado de datos relacionales los diagramas de clases son los más adecuados. Existe una correspondencia entre las entidades y las clases en estos diagramas, y entre las relaciones y las asociaciones genéricas en diagramas de clases.



Debido a la diversidad de opciones que tenemos para los datos no estructurados, es un poco más difícil hablar de una notación o lenguaje específico; sin embargo, la representación basada en estructuras de **árboles**, que tienen un **nodo raíz** y **sub nodos** asociados de forma jerárquica a este

nodo, es una forma adecuada para la representación de datos **no relacionales orientados a documentos JSON**. Los datos no relacionales orientados a documentos presentan como principal característica la agrupación de documentos en forma jerárquica. En este caso no tenemos una herramienta de software para dibujar los árboles, por el contrario, cualquier software que soporte las representaciones de estructuras de árboles se puede usar para documentar datos NoSQL orientados a documentos.

INTRODUCCIÓN A LAS BASES DE DATOS

El curso de Introducción a las Base de datos ha sido un viaje a través de los fundamentos de estos repositorios de datos. Comenzaste por identificar los datos en un contexto relevante para la solución de tu problema y su posterior agrupación en entidades y sus atributos y tipos.

Después de conocer la arquitectura de un sistema de base de datos, revisamos los paradigmas más comunes, agrupados en Relacionales y No Relacionales, también conocidos como SQL y NoSQL, respectivamente.

No sólo pudiste distinguir entre estos tipos de base de datos, sino que también manipulaste las mismas usando herramientas y estructuras de manejadores, tales como MySQL, para bases de datos relacionales, y MongoDB, para las no relacionales.

Para finalizar, exploramos las formas de documentar tus datos usando diagramas entidad/relación para los datos SQL y las estructuras jerárquicas basadas en árboles de documentos para los datos no relacionales tipo JSON.

Luego de haber transitado por este camino introductorio de la bases de datos has desarrollado la capacidad de:

- Interpretar los tipos de bases de datos para establecer las relaciones entre entidades de un sistema.
- Crear la estructura de base de datos que se adapta a la solución basado en mejores prácticas

- Formular diagramas de bases de datos usando notaciones estándares del mercado
- Diferenciar los sistema manejadores de base de datos con la finalidad de familiarizarse con los productos existentes en el mercado.

Ahora estás listo para continuar tu travesía hacia conceptos más avanzados de las bases de datos en los próximos cursos. Pero primero, te dejamos un resumen de lo que hemos cubierto en la Introducción a las Bases de Datos.