

> lenguaje de consulta estructurada
Structured Query Language

Introducción a sql

SEMANA 6



SQL INTRODUCCIÓN

Introducción a la Gestión de Bases de Dato

En la actualidad, el acceso rápido y confiable a la información es crucial para el funcionamiento de numerosas organizaciones y aplicaciones. Las bases de datos se han convertido en la columna vertebral de diversas soluciones tecnológicas, empleándose en una amplia gama de campos, como empresas, instituciones educativas, la industria médica, las finanzas y otros sectores.

Las bases de datos son sistemas estructurados y organizados que nos permiten almacenar, gestionar y recuperar datos eficientemente. Los datos almacenados pueden ser de naturaleza diversa: desde información sobre clientes y productos, hasta registros de transacciones, resultados de experimentos científicos y cualquier otra información que necesitemos almacenar y consultar.

En este contexto, es fundamental comprender los conceptos fundamentales de las bases de datos relacionales, comenzando por entender los distintos tipos

de modelos de datos y cómo representar adecuadamente la información. Aprenderás los principios básicos del diseño de bases de datos, incluyendo la creación de tablas, definición de relaciones y establecimiento de claves primarias y foráneas (Pk y Fk).

Además, exploraremos el lenguaje de consulta más utilizado en la gestión de bases de datos, el famoso SQL (Structured Query Language). Con él, podrás realizar consultas sofisticadas para extraer información específica de las bases de datos.

Abordaremos temas como la normalización, que nos permitirá optimizar nuestras bases de datos y evitar redundancias innecesarias, y la seguridad de los datos, un aspecto crítico para garantizar la privacidad y confidencialidad de la información almacenada.

Prepárate para explorar el apasionante mundo de las bases de datos y descubrir cómo estas herramientas pueden potenciar tus proyectos y habilidades como profesional de la tecnología.

LA IMPORTANCIA DE LAS BASES DE DATOS



Primero, recolectamos información relevante, tales como hechos y eventos. Una vez recopilados, los datos se almacenan en un registro donde se valida la veracidad y precisión de los mismos para evitar errores o inconsistencias en el procesamiento. Posteriormente, los datos se procesan, ya sea mediante la categorización o clasificación, para generar informes o reportes y proporcionar la información necesaria cuando sea requerida.

Aunque los datos pueden parecer sin sentido al principio, el tratamiento adecuado les da un propósito para cumplir con los objetivos de información que se buscan alcanzar.

¿Cómo se convierten Datos en Información?

1 - La conversión de datos en información es un proceso o un conjunto de tareas (generalmente llamado proceso de transformación) relacionadas entre sí en formas lógicas y ejecutadas con el fin de producir un resultado (información útil).

2 - El proceso para definir relaciones entre datos requiere de conocimiento.

3 - El conocimiento es la apreciación y compresión de un conjunto de información y de la utilidad que puede extraerse de ella en beneficio de una tarea e

4 - En algunos casos los datos se organizan o procesan en forma mental o manual.

5 - No importa de dónde proceden los datos o cómo se los procesa sino lo útil y valiosos que son.

Por ej:



Como se puede apreciar, en este caso específico, se almacenan en una base de datos tanto los datos de acceso del usuario, como sus datos personales completos. Si bien no es relevante en este ejemplo conocer el origen de dicha información, es importante destacar que el tratamiento que se le da a los datos es fundamental para el correcto funcionamiento de un sistema de inicio de sesión. A través de la comparación de los datos almacenados con los que el usuario provee al momento de iniciar sesión, es posible determinar qué información brindarle al usuario.

Datos vs Información

Para ver esta diferencia vamos a remontarnos a los conceptos básicos:

Datos

Un dato es una representación simbólica de información específica utilizada para describir hechos, conceptos o ideas. Es una unidad fundamental de información que se puede almacenar, procesar y analizar dentro de una base de datos. A continuación, se presentan algunos puntos esenciales sobre los datos:

1 - Significado: Los datos tienen un significado y representan (luego de procesarlos) información sobre un objeto, evento o entidad en el mundo real. Por ejemplo, el nombre de una persona, su dirección o su número de teléfono son datos que representan información sobre esa persona.

2 - Tipos de datos: Los datos pueden ser de diferentes tipos, como texto, números, fechas, imágenes o videos. Cada tipo de dato tiene características y propiedades específicas que determinan cómo se almacena y manipula.

3 - Estructura: Los datos se pueden organizar en diversas estructuras dependiendo del modelo de base de datos utilizado. En las bases de datos relacionales, los datos se organizan en campos, registros y tablas. Los campos representan atributos individuales, los registros contienen conjuntos de campos relacionados y las tablas agrupan registros con la misma estructura. Por otro lado, en las bases de datos jerárquicas, los datos se organizan en una estructura en forma de árbol.

4 - Precisión: Los datos deben ser precisos y correctos para que sean útiles y confiables. La precisión implica la ausencia de errores, inconsistencias o ambigüedades en los datos.

5 - Integridad: La integridad de los datos se refiere a su exactitud y coherencia a lo largo del tiempo y en diferentes sistemas o procesos. Los mecanismos de integridad aseguran que los datos se mantengan consistentes y no se corrompan.

6 - Acceso y manipulación: Los datos deben poder ser accedidos y manipulados para su uso. Los sistemas de gestión de bases de datos (DBMS) proporcionan herramientas y lenguajes de consulta, como SQL, que permiten buscar, modificar, agregar y eliminar datos de manera eficiente.

7 - Privacidad y seguridad: Los datos a menudo contienen información sensible y personal. La privacidad y seguridad de los datos son aspectos cruciales, y deben aplicarse medidas para proteger la confidencialidad, integridad y disponibilidad de la información.

8 - Interrelaciones: En una base de datos, los datos suelen estar vinculados entre sí, estableciendo relaciones mediante claves primarias y foráneas. Esto permite establecer asociaciones y vínculos entre diferentes conjuntos de datos.

9 - Utilidad: Los datos son valiosos cuando se utilizan para tomar decisiones informadas, generar conocimiento o proporcionar información relevante. La calidad y utilidad de los datos determinan en gran medida su impacto y utilidad en diferentes contextos.

8 - Interrelaciones: En una base de datos, los datos suelen estar vinculados entre sí, estableciendo relaciones mediante claves primarias y foráneas. Esto permite establecer asociaciones y vínculos entre diferentes conjuntos de datos.

9 - Utilidad: Los datos son valiosos cuando se utilizan para tomar decisiones informadas, generar conocimiento o proporcionar información relevante. La calidad y utilidad de los datos determinan en gran medida su impacto y utilidad en diferentes contextos.

10 - Evolución: Los datos son susceptibles a cambios con el tiempo a medida que se actualizan, agregan o eliminan. El seguimiento y control de los cambios son aspectos importantes para mantener la integridad y el historial de la información.

Estos puntos clave sobre los datos proporcionan una base sólida para comprender la importancia de una gestión adecuada de las bases de datos y cómo los datos pueden convertirse en activos valiosos para cualquier organización o proyecto.

Las bases de datos relacionales se utilizan comúnmente cuando se requiere una alta estructura y coherencia en los datos, especialmente cuando hay relaciones complejas entre ellos. Algunos casos típicos en los que se utilizan bases de datos relacionales incluyen:

Sistemas de gestión de bases de datos empresariales: Grandes organizaciones como bancos, compañías de seguros o empresas de comercio electrónico, utilizan bases de datos relacionales para gestionar datos transaccionales, información de clientes, inventarios y otros aspectos clave de su negocio.

Aplicaciones web y móviles: Muchas aplicaciones web y móviles utilizan bases de datos relacionales para almacenar y gestionar datos de usuarios, contenido dinámico, comentarios y otras interacciones. Estas bases de datos ofrecen una estructura sólida y permiten consultas complejas y relaciones entre tablas.

Grandes volúmenes de datos no estructurados: Si se trabaja con datos no estructurados, como textos, imágenes, videos o datos geoespaciales, las bases de datos no relacionales, como las bases de datos de documentos o las bases de datos de grafos, pueden ser más apropiadas. Estas bases de datos permiten una mayor flexibilidad y escalabilidad al manejar grandes cantidades de datos diversos.

Aplicaciones con necesidades de rendimiento y escalabilidad: Algunas aplicaciones con altos requisitos de rendimiento y escalabilidad, como redes sociales, plataformas de juegos en línea o sistemas de análisis de big data, pueden beneficiarse de las bases de datos no relacionales. Estas bases de datos están diseñadas para distribuir y procesar grandes volúmenes de datos en paralelo, lo que les permite escalar eficientemente a medida que aumenta la demanda.

Algunos ejemplos de bases de datos relacionales populares incluyen Oracle, MySQL, PostgreSQL y Microsoft SQL Server. Por otro lado, ejemplos de bases de datos no relacionales o jerárquicas son MongoDB, CouchDB, Redis y Neo4j, esta última especializada en bases de datos de grafos.

información

La información es el resultado del procesamiento y análisis de datos, lo cual otorga significado y contexto a los hechos o eventos. Se refiere a los datos organizados y estructurados de manera que sean útiles y comprensibles para los usuarios. A continuación, se presentan algunos aspectos fundamentales sobre la información:

Significado: La información es significativa y relevante para los usuarios, ya que proporciona conocimiento, comprensión y perspectiva sobre un tema específico, trascendiendo los datos simples.

Interpretación: La información implica la interpretación y análisis de los datos disponibles. Los datos por sí solos no tienen valor significativo, hasta que se les da un contexto y se les da sentido a través de procesos de interpretación y análisis.

Relevancia: La información es relevante para un propósito específico o para la toma de decisiones, y debe estar relacionada directamente con las necesidades y objetivos de quienes la utilizan.

Calidad: La información debe ser precisa, confiable y libre de errores. La calidad de la información se refiere a su exactitud, integridad y actualidad, así como a la confiabilidad de la fuente de la cual se obtiene.

Accesibilidad: La información debe estar disponible y accesible para aquellos que la necesitan. Los sistemas de gestión de bases de datos y otros sistemas de información proporcionan mecanismos para almacenar, recuperar y compartir información de manera eficiente.

Utilidad: La información es útil cuando se aplica para tomar decisiones informadas, resolver problemas o generar conocimiento.

Su valor radica en su capacidad para ayudar a los usuarios a comprender mejor una situación, predecir resultados o guiar acciones.

Confidencialidad: La información sensible y confidencial debe ser protegida para evitar su divulgación no autorizada. Esto implica la implementación de medidas de seguridad y la adhesión a políticas de privacidad y protección de datos.

Actualización: La información puede cambiar con el tiempo debido a la actualización de los datos subyacentes. Es importante mantener la información actualizada y reflejar los cambios relevantes en los datos para garantizar su relevancia y precisión continua.

Presentación: La información se presenta de manera visual, textual o en otros formatos comprensibles para los usuarios. La presentación efectiva de la información implica la selección de formatos apropiados y la comunicación clara de los resultados del análisis.

Valor agregado: La información puede tener un valor agregado cuando se combina, analiza o se le da contexto adicional. El procesamiento de los datos para convertirlos en información brinda la oportunidad de descubrir patrones, tendencias o relaciones que no son evidentes en los datos individuales.

Estos puntos fundamentales destacan la importancia de la información como un recurso valioso en la toma de decisiones, la generación de conocimiento y el logro de los objetivos organizacionales.

Sistemas de información

Los sistemas de información son conjuntos de componentes interrelacionados que trabajan juntos para recopilar, procesar, almacenar y distribuir información con el fin de apoyar la toma de decisiones, la coordinación, el control y el análisis dentro de una organización. Estos sistemas involucran tanto aspectos tecnológicos como humanos, y su objetivo principal es mejorar la eficiencia y efectividad de las operaciones organizativas. Aquí hay algunos aspectos clave relacionados con los sistemas de información:

1.Componentes: Los sistemas de información consisten en diferentes componentes interdependientes. Estos componentes incluyen hardware (como computadoras y dispositivos de almacenamiento), software (aplicaciones y programas informáticos), datos (información almacenada y procesada) y personas (usuarios y profesionales encargados de utilizar y administrar el sistema).

2.Procesamiento de datos: Los sistemas de información recopilan datos crudos y los procesan para convertirlos en información significativa y útil. Esto implica actividades como la entrada de datos, el almacenamiento, la recuperación, la manipulación y la presentación de la información procesada de manera comprensible.

3.Tipos de sistemas de información: Hay varios tipos de sistemas de información utilizados en diferentes áreas y niveles organizativos.



ERP: significa Enterprise Resource Planning (Planificación de Recursos Empresariales) y se refiere a un sistema de software integral diseñado para integrar y gestionar los diferentes procesos y funciones de una organización en un solo sistema. Un sistema ERP centraliza y automatiza actividades como la gestión de inventarios, compras, ventas, contabilidad, recursos humanos y producción, entre otros. Proporciona una visión holística de la organización y permite un flujo de información eficiente entre los departamentos, lo que facilita la toma de decisiones informadas y mejora la eficiencia operativa.

OAS: significa Office Automation System (Sistema de Automatización de Oficina) y se refiere a un conjunto de herramientas y aplicaciones de software que se utilizan para mejorar y automatizar las tareas y procesos relacionados con la oficina. Estos sistemas abarcan funciones como la gestión de documentos, la comunicación interna y externa, el flujo de trabajo, la programación de reuniones y la colaboración en equipo. Los OAS ayudan a optimizar la productividad y la eficiencia en el entorno de la oficina, al tiempo que mejoran la comunicación y la coordinación entre los miembros del equipo.

Sistemas de Información Ejecutiva/Sistemas de Soporte a la Toma de Decisiones (SE/SSG): Estos sistemas están diseñados para ayudar a los altos directivos y ejecutivos en la toma de decisiones estratégicas. Proporcionan información resumida y análisis de datos clave de la organización, ayudando a identificar tendencias, patrones y escenarios futuros.

Sistemas de Información Gerencial (MIS)/Sistemas de Soporte a la Decisión (DSS): Los MIS se enfocan en proporcionar información y análisis para ayudar a los gerentes en la toma de decisiones operativas y tácticas. Los DSS son sistemas más flexibles que brindan herramientas y modelos interactivos para ayudar en la resolución de problemas y la toma de decisiones en tiempo real.

Sistemas de Gestión del Conocimiento (KWS): Estos sistemas se centran en capturar, organizar y gestionar el conocimiento y la experiencia de los empleados dentro de una organización. Ayudan a compartir y acceder a información relevante, mejores prácticas y lecciones aprendidas, fomentando la colaboración y el aprendizaje en la organización.

Sistemas de Procesamiento de Transacciones (TPS): Los TPS se utilizan para procesar y gestionar transacciones diarias y rutinarias de una organización. Estos sistemas automatizan y registran actividades como ventas, compras, inventario y nóminas. Proporcionan información precisa y actualizada para respaldar la toma de decisiones operativas y generar informes financieros.

Cada tipo de sistema de información tiene su propósito y enfoque específicos dentro de una organización, y su implementación depende de las necesidades y objetivos particulares de la misma.

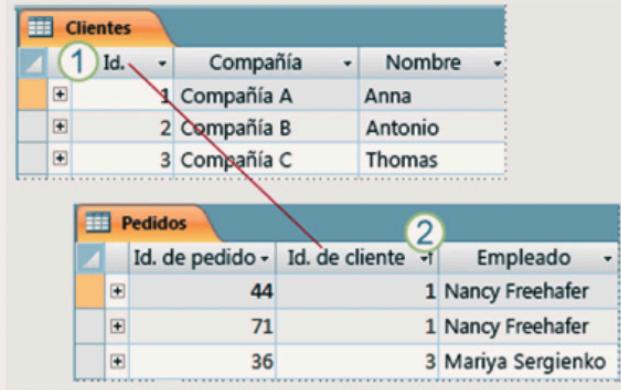
BASES DE DATOS

Es evidente que las bases de datos desempeñan una función esencial en la administración de información en las organizaciones. A través del almacenamiento, recuperación, actualización y análisis eficiente de datos, se facilita la toma de decisiones, la generación de informes y el desarrollo de aplicaciones. La correcta concepción y gestión de una base de datos resultan críticas para asegurar la integridad, la seguridad y el rendimiento de los datos en una organización.

Ilustramos las diferencias entre una base de datos SQL o Relacional y una base de datos NoSQL o Jerárquica de forma gráfica en el siguiente ejemplo.

Base de Datos Relacional (SQL)

Si es almacenada en forma tabular entonces es llamada **base de datos relacional**.

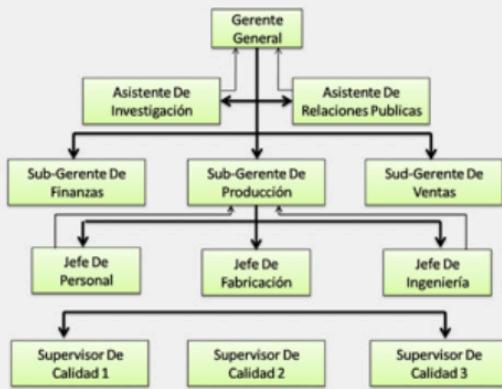


Clientes		
	Id.	Compañía
1	1	Compañía A
	2	Compañía B
	3	Compañía C

Pedidos		
	Id. de pedido	Id. de cliente
1	44	1 Nancy Freehafer
	71	1 Nancy Freehafer
	36	3 Mariya Sergienko

Base de Datos Jerárquica (NoSQL)

Si los datos están organizados en forma de estructura de árbol, es llamada **base de datos jerárquica**.

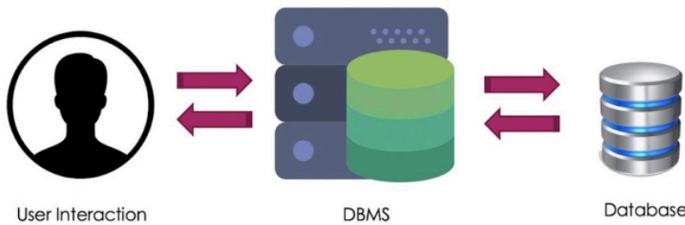


Base de datos Relacional (SQL)

Para abordar este tema, vamos a hablar sobre los Sistemas de Gestión de Bases de Datos (SGBD).

Sistema de gestión de base de datos

Las aplicaciones con las que interactúa un usuario normalmente no tienen acceso directo a una Base de Datos, para ello se interpone entre ambas un nuevo nivel de software que llamaremos Sistema de Gestión de Base de Datos (SGBD, o DBMS en inglés).



El SGBD consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos.

El SGBD provee generalmente una aplicación especial llamada lenguaje de consulta (query language) que permite a los usuarios finales realizar consultas a la Base de Datos.

Funciones de un sgbd

- Creación y definición de la base de datos:** Los SGBD permiten crear y definir la estructura de la base de datos, especificando las tablas, campos, tipos de datos y relaciones entre ellos. Proporcionan herramientas para diseñar esquemas de bases de datos y establecer restricciones de integridad para garantizar la consistencia de los datos.
- Manipulación de datos:** Los SGBD permiten realizar operaciones de inserción, actualización, eliminación y consulta de datos. A través de lenguajes de consulta, como SQL, los usuarios pueden recuperar información de la base de datos y realizar cambios en los datos de manera eficiente.
- Control de la concurrencia y la seguridad:** Los SGBD gestionan el acceso concurrente a la base de datos, asegurando que varias personas o aplicaciones puedan acceder a los datos de manera segura y sin interferencias. Además, proporcionan mecanismos de seguridad para controlar el acceso a la base de datos y proteger la información sensible.
- Optimización del rendimiento:** Los SGBD implementan técnicas de optimización para mejorar el rendimiento de las consultas y las operaciones en la base de datos. Utilizan índices, estrategias de almacenamiento eficientes y algoritmos de optimización para acelerar las consultas y minimizar el tiempo de respuesta.
- Recuperación y respaldo de datos:** Los SGBD incluyen mecanismos de recuperación de datos en caso de fallos o errores, asegurando la integridad de la base de datos. También ofrecen funcionalidades para realizar copias de seguridad y restaurar datos, permitiendo recuperar la información en caso de pérdida o corrupción.

Algunos ejemplos de Sistemas de Gestión de Bases de Datos Relacionales (RDBMS) y sus respectivas bases de datos relacionales:

MySQL: es un sistema de gestión de bases de datos relacionales de código abierto ampliamente utilizado. Es conocido por su rendimiento, confiabilidad y facilidad de uso. Algunos ejemplos de bases de datos desarrolladas con MySQL son WordPress, Joomla y Drupal o Moodle (que es la plataforma que usas para acceder a esta información en el Cenit).

Oracle Database: es un RDBMS líder en el mercado y es utilizado por muchas grandes empresas. Es conocido por su escalabilidad, seguridad y robustez. Ejemplos de aplicaciones basadas en Oracle Database incluyen sistemas de gestión empresarial (ERP), sistemas de recursos humanos y sistemas de gestión de relaciones con los clientes (CRM).

Microsoft SQL Server: es un sistema de gestión de bases de datos relacionales desarrollado por Microsoft. Es ampliamente utilizado en entornos empresariales y ofrece una amplia gama de características y herramientas. Algunos ejemplos de aplicaciones que utilizan Microsoft SQL Server son sistemas de gestión de inventario, sistemas de seguimiento de pedidos y sistemas de gestión de relaciones con los clientes.

PostgreSQL: es un sistema de gestión de bases de datos relacional de código abierto y altamente extensible. Es conocido por su capacidad de manejar cargas de trabajo complejas y por su enfoque en la conformidad con los estándares. PostgreSQL es utilizado en una amplia gama de aplicaciones, desde pequeñas hasta grandes empresas, y se destaca en áreas como la geolocalización, el análisis de datos y la gestión de contenidos.

Estos son solo algunos ejemplos de bases de datos relacionales y los sistemas de gestión de bases de datos que los respaldan. Cada uno tiene sus propias características y fortalezas, por lo que es importante evaluar las necesidades específicas antes de elegir el sistema de gestión de bases de datos adecuado para un proyecto o aplicación.

En este curso usaremos MySQL como nuestro RDBMS.

Nivel de Abstracción Sistema de Gestión de Base de Datos

El nivel de abstracción en un Sistema de Gestión de Base de Datos (SGBD) se refiere a las diferentes capas o niveles de representación y operación que se utilizan para interactuar con la base de datos. Estos niveles están diseñados para facilitar el acceso y la manipulación de los datos, brindando a los usuarios una interfaz adecuada para sus necesidades específicas. Los tres niveles de abstracción comunes en un SGBD son:

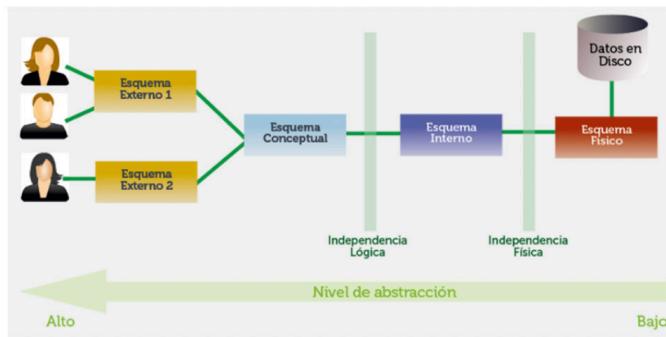
Nivel físico (o básico): El nivel físico es el nivel más bajo de abstracción y se ocupa de la forma en que los datos se almacenan físicamente en el disco o en otros medios de almacenamiento. Define cómo se organizan los datos en bloques, páginas o registros, y cómo se accede a ellos. En este nivel, se consideran aspectos como la estructura del archivo, el acceso a los datos y la optimización del almacenamiento. Los usuarios y aplicaciones típicamente no interactúan directamente con este nivel.

Nivel lógico (o conceptual): El nivel lógico proporciona una descripción más abstracta de la base de datos y se enfoca en cómo se estructuran y organizan los datos a nivel conceptual. En este nivel, los datos se

representan mediante modelos de datos, como el modelo relacional, el modelo de entidad-relación, el modelo de objetos, entre otros. Los usuarios interactúan con la base de datos a través de consultas y comandos que se definen en función de este nivel lógico. El SGBD se encarga de traducir estas consultas y comandos en instrucciones físicas para acceder a los datos en el nivel físico.

Nivel de vista o nivel de usuario

(o externo): El nivel de vista o nivel de usuario es el nivel más alto de abstracción y es el que los usuarios finales y las aplicaciones utilizan para interactuar con la base de datos. En este nivel, los usuarios definen consultas y realizan operaciones sobre la base de datos utilizando un lenguaje de consulta, como SQL (Structured Query Language). Los usuarios ven una representación simplificada y personalizada de la base de datos, que puede incluir vistas, informes y formularios adaptados a sus necesidades específicas. El nivel de vista oculta la complejidad de la estructura y organización de la base de datos subyacente.



En esta Figura se observa la distancia que poseen los usuarios de la base de datos respecto a la realidad física de la base de datos (representada con el cilindro). La física son los datos en crudo, es decir en formato binario dentro del disco o discos que los contienen. El esquema físico es el que se realiza pensando más en esa realidad y los esquemas externos los que se crean pensando en la visión de los usuarios.

Independencia de datos

Como se puede observar, en la imagen hay dos columnas que representan niveles de Independencia de Datos.

La Independencia de Datos hace referencia a la capacidad de actualizar la definición del esquema de un nivel sin que afecte a la definición de esquema.

Existen **2 niveles** de independencia de datos:

Independencia física de datos: La da la barrera entre el esquema físico y el interno e indica que el esquema interno es independiente del hardware. Estas modificaciones suelen ser necesarias para mejorar el funcionamiento (ej: modificaciones en los métodos de acceso, cambios en la estructura de almacenamiento, etc).

Independencia lógica de datos: Los esquemas de los niveles conceptual y externo son independientes del software concreto de base de datos que usemos; no dependen en absoluto de él (ej: agregar o quitar entidades).

La independencia lógica es más difícil de proporcionar que la independencia de datos físicos, ya que los programas de aplicación son fuertemente dependientes de la estructura lógica de los datos a los que acceden.

Lenguaje de Base de Datos



Lenguaje de Definición de Datos:

Un esquema de BD o la estructura de la BD se especifica mediante un conjunto de definiciones expresadas mediante un lenguaje especial llamado lenguaje de definición de datos (DDL). El resultado de la compilación de las instrucciones del LDD es un conjunto de tablas que se almacenan en un archivo especial llamado diccionario de datos.

Lenguaje de manipulación de datos:

La manipulación se refiere a:

- Inserción de información nueva en la Base de Datos. **(C)**
- Recuperación de la información almacenada en la Base de Datos. **(R)**
- Modificación de información almacenada en la Base de Datos. **(U)**

Eliminación de información en la Base de Datos. **(D)**

CRUD por sus siglas en inglés

(C) - Create - Crear

(R) - Read - Leer

(U) - Update - Actualizar

(D) - Delete - Eliminar

A lo largo de este curso escucharemos o haremos referencia acerca de "realizar una consulta a la BD".

Una consulta es una instrucción de solicitud para recuperar información de la BD.

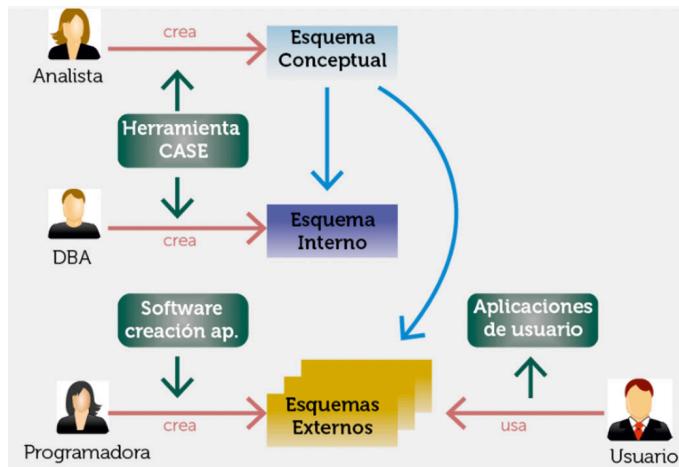
Arquitecto/a de Base de Datos (Database Architect):

Similar al rol anterior, aunque el trabajo está más orientado a la solución de una base de datos.

Administrador/a de Base de Datos (Database Administrator - DBA): Es el responsable del mantenimiento, desempeño, integridad y seguridad de una base de datos.

Desarrollador/a de Aplicaciones (Application Developer): Es la persona encargada del desarrollo de aplicaciones que acceden a la base de datos.

Recursos humanos de las bases de datos



Arquitecto/a de Datos (Data Architect): Es responsable de diseñar una arquitectura que soporte la actual o futuras necesidades de administración de los datos de la organización.

> Modelado de base de datos

Modelado de base de datos

INTRODUCCIÓN

Modelado de datos

El modelado de datos es el proceso de representar y organizar la estructura y relaciones de los datos en un sistema de información mediante modelos de datos. El objetivo principal del modelado de datos es comprender y visualizar cómo se organizan los datos y cómo interactúan entre sí. Esto proporciona una base sólida para el diseño y desarrollo de bases de datos eficientes y precisas.

	carro_id	tipo	color	tienda	precio
1	1	5 negro	1	203000	
2	2	2 amarillo	1	150	
3	3	3 verde	3	14800	
4	4	4 negro	2	10200	
5	5	1 verde	4	82399	
6	6	1 azul	4	22374	

	tipo_id	marca_id	nombre
1	1	3 tipo e	
2	2	2 1118	
3	3	4 golf	
4	4	4 polo	
5	5	1 quattroporte	

	marca_id	nombre	pais_de_origen
1	1	maserati	italia
2	2	lada	rusia
3	3	jaguar	reino unido
4	4	volkswagen	alemania

	tienda_id	nombre	calle	codigo_postal	telefono
1	1	amsterdam south	churchill	1079HA	20373
2	2	amsterdam west	mercator	1056RT	20838
3	3	amsterdam north	buiksloterstraat	1031AB	20387
4	4	the hague	neherstrem	2491JJ	70387



SEMANA 6

En la imágenes anteriores vemos la representación de una Base de Datos y te mencionamos algunas cosas importantes:

- Cada base de datos se compone de una o más tablas que guarda un conjunto de datos.
- Cada tabla tiene una o más columnas y filas.
- Las columnas guardan una parte de la información sobre cada elemento que queremos guardar en la tabla, cada fila de la tabla conforma un registro.

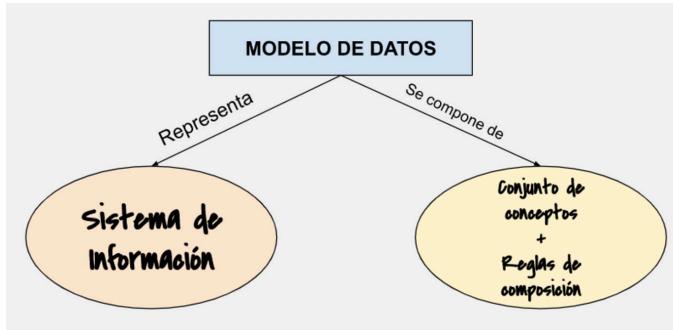
No importa si nuestra base de datos tiene sólo 20 registros, o algunos cuantos miles, es importante asegurarnos que nuestra base de datos está correctamente diseñada para que tenga eficiencia y que se pueda seguir utilizando por largo del tiempo.

El diseño de bases de datos es el proceso por el que se determina la organización de una base de datos, incluidos su estructura, contenido y las aplicaciones que se han de desarrollar.

Para ello necesitamos conocer:

¿Qué es un modelo de datos?

Es muy importante seleccionar el modelo de datos a utilizar, ya que ayuda a los equipos de desarrollo de software a diseñar y organizar datos asociados al proyecto de software.



Para comprender los tipos de Modelo de Datos debemos empezar sabiendo que el diseño de una base de datos se descompone en **diseño conceptual**, **diseño lógico** y **diseño físico**.

Diseño conceptual

Parte de las especificaciones de requisitos de usuario y su resultado es el esquema conceptual de la base de datos.

¿Qué es un esquema conceptual?

Es una descripción de alto nivel de la estructura de la base de datos, independientemente del DBMS que se vaya a utilizar para manipularla.

El objetivo del diseño conceptual es describir los datos de la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar estos datos.

Diseño lógico

Parte del esquema conceptual y da como resultado un esquema lógico.

¿Qué es un esquema lógico?

Es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de DBMS.

Un modelo lógico es un lenguaje usado para especificar esquemas lógicos (modelo relacional, modelo de red, etc.).

El diseño lógico depende del tipo de DBMS que se vaya a utilizar, no depende del producto concreto.

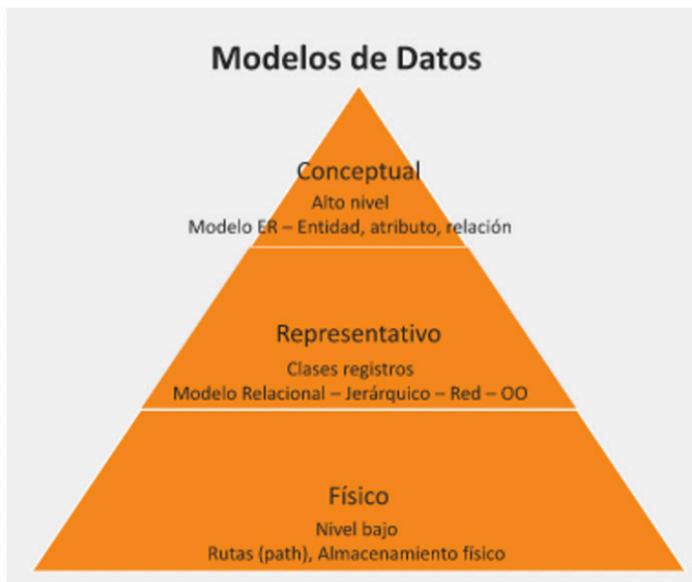
Diseño físico

Parte del esquema lógico y da como resultado un esquema físico.

¿Qué es un esquema físico?

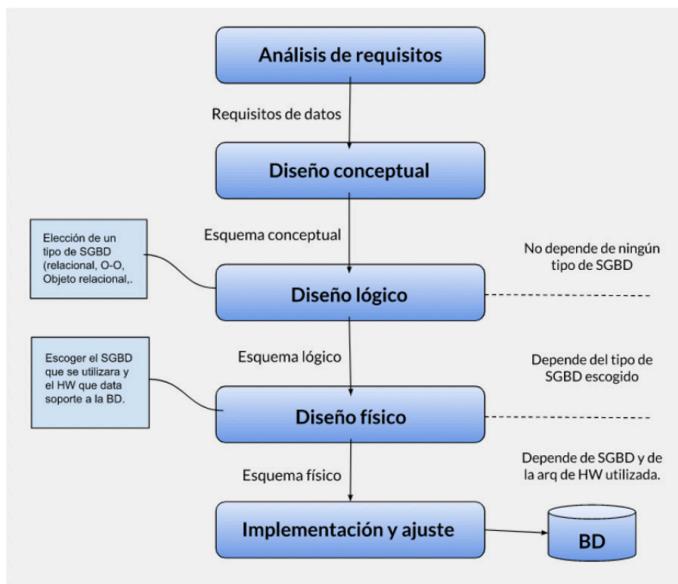
Es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos.

Por ello, el diseño físico depende del DBMS.



Proceso de Diseño de la Base de datos

Se muestra a continuación una descripción simplificada del proceso de diseño de la base de datos.



1 - Obtención y análisis de requisitos:

Durante este paso los diseñadores entrevistan a los futuros usuarios de la base de datos para entender y documentar sus requisitos de datos.

El resultado de este paso será un conjunto de requisitos del usuario, redactado de forma concisa. Estos requisitos deben especificarse de la forma más detallada y completa posible.

2 - Esquema conceptual.

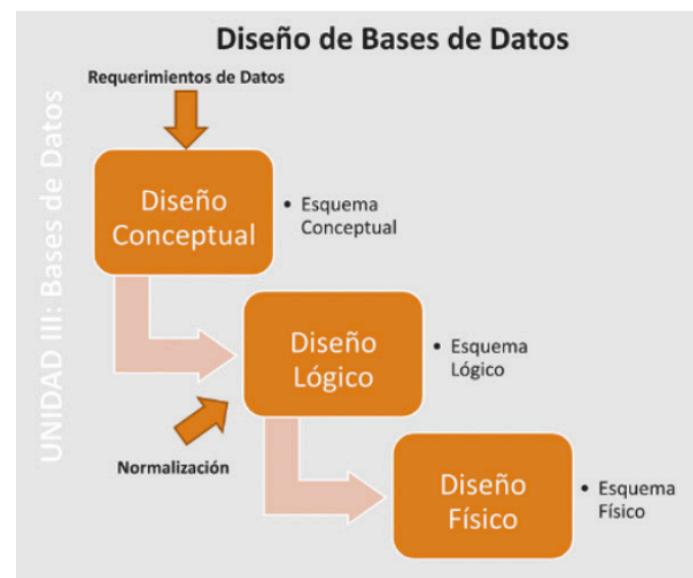
Se crea un esquema conceptual para la base de datos mediante un modelo conceptual de datos de alto nivel. Este paso se denomina diseño conceptual. El esquema conceptual es una descripción concisa de los requisitos de información de los usuarios, y contiene descripciones detalladas de los tipos de entidad, vínculos y restricciones representados según el modelo conceptual de datos usado. Puestos que estos conceptos no incluyen detalles de implementación, suelen ser fáciles de entender y pueden servir para comunicarse con usuarios no técnicos.

El esquema conceptual de alto nivel también puede servir como referencia para asegurarse de satisfacer todos los requerimientos de los usuarios y de que no haya conflictos entre dichos requisitos.

3 - Una vez diseñado el esquema conceptual o durante dicho proceso, es posible utilizar las operaciones básicas del modelo de datos para especificar operaciones de usuario de alto nivel identificadas durante el análisis funcional.

A partir de allí se debe usar un SGBD para implementar la base de datos. Esto se logra transformando el esquema conceptual del modelo usado al modelo de datos de implementación. Este paso se llama diseño lógico o transformación del modelo de datos, y su resultado es un esquema de la base de datos en el modelo de datos que se usará para la implementación de la base de datos.

4 - El paso final es la fase de diseño físico, durante la cual se especifican las estructuras de almacenamiento internas, los caminos de acceso y la organización de ficheros de la base de datos. Por ejemplo, en el curso desarrollaremos una aplicación y utilizaremos un SGBD para trabajar con los datos (MySQL Workbench).



> MER - modelo entidad relación

Modelado de base de datos MER



MER

MODELADO - ENTIDAD - RELACIÓN

Denominado por sus siglas como: MER, este modelo representa a la realidad a través de entidades, que son objetos que existen y que se distinguen de otros por sus características.

¿CUÁLES SON LOS ELEMENTOS DE UN MODELO?

1. ENTIDAD
2. ATRIBUTOS
3. RELACIÓN
- A. GRADO
- B. CARDINALIDAD
- C. ATRIBUTOS PROPIOS DE UNA RELACIÓN
- D. CLAVE DE UNA RELACIÓN
4. CLAVES

ENTIDAD

Una entidad es un objeto del mundo real que puede distinguirse de otros objetos. Una entidad puede ser un objeto con existencia física (una persona, un automóvil, una casa, un empleado) o un objeto con existencia conceptual (una empresa, un puesto de trabajo, un curso universitario, una cuenta de cliente).

ATRIBUTO

Cada entidad tiene propiedades o características específicas llamadas atributos que la describen. Por ejemplo, una entidad Cliente puede describirse por su nombre, fecha de nacimiento, dirección, dni y monto adeudado.

Atributos compuestos o simples

Los atributos compuestos se pueden dividir en componentes más pequeños. Por ejemplo, para una entidad Cliente el atributo Dirección en realidad puede ofrecer información sobre Calle, Número, Localidad, etc. En cambio, los atributos simples no pueden subdividirse, por ejemplo, Nombre, Edad.

Atributos monovaluados o multivaluados

Los atributos monovaluados son aquellos que pueden tener un único valor por entidad, por ejemplo, para la entidad Cliente el atributo Género es monovaluado.

En cambio, un atributo que puede tener un conjunto de valores para la misma entidad es un atributo multivaluado, por ejemplo, para la entidad Cliente el atributo Profesión, ya que la persona puede tener varias Profesiones.

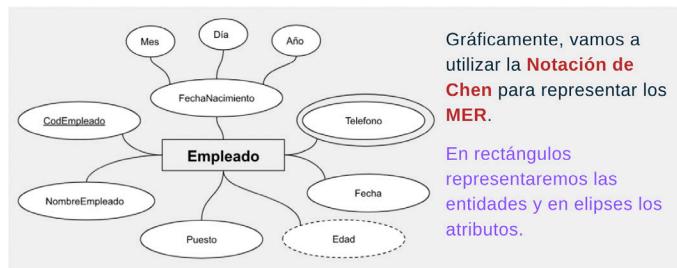
Atributos almacenados o derivados

En algunos casos, los valores de dos o más atributos están relacionados, por ejemplo, los atributos Edad y Fecha de Nacimiento de una persona. Para un registro particular,

el valor de la Edad puede estar determinado por la fecha actual y la fecha de nacimiento de la persona. Entonces el atributo Edad recibe el nombre de atributo derivado y se dice que es derivado del atributo Fecha de Nacimiento, el cual recibirá el nombre de atributo almacenado.

Además, para cada conjunto de entidades se escoge un atributo clave.

UNA CLAVE ES UN CONJUNTO MÍNIMO DE ATRIBUTOS CUYOS VALORES IDENTIFICAN DE MANERA UNÍVOCAMENTE A CADA ENTIDAD DEL CONJUNTO. PUEDE HABER MÁS DE UNA CLAVE CANDIDATA; EN ESE CASO, SE ESCOGERÁ UNA DE ELLAS COMO CLAVE PRINCIPAL.



Los **atributos compuestos** se van a representar incluyendo los atributos simples en los que se descomponen, como el caso de FechaNacimiento.

Los **atributos multivaluados** se van a representar con elipses dobles, como el caso de Teléfono (ya que cada empleado puede tener más de uno).

Los **atributos derivados** o calculados como el ejemplo de la edad, se van a representar con elipses de línea punteada.

Los **atributos claves** serán subrayados en el diagrama, como es en este caso CodEmpleado.

RELACIÓN

Una relación es una asociación entre dos o más entidades.

Por ejemplo, puede que se tenga que modelar una relación en la cual un empleado haya trabajado en un ticket determinado.



En el conjunto de relaciones trabajó, cada relación indica un trabajo efectuado por ese empleado. Se puede observar que puede que, varios conjuntos de relaciones impliquen a los mismos conjuntos de entidades.

Veremos ahora las partes que conforman las relaciones.

a) Grado

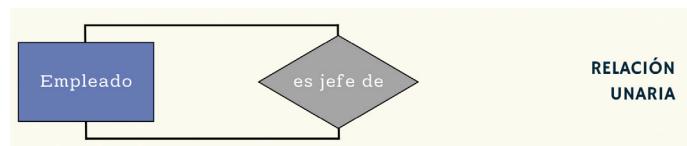
Se refiere al número de entidades que participan en una relación.

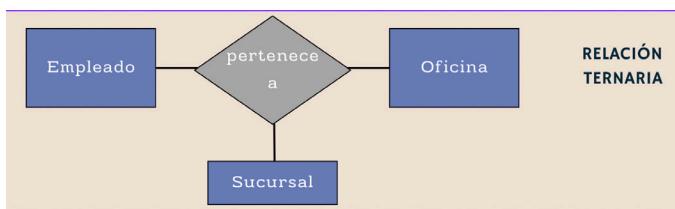
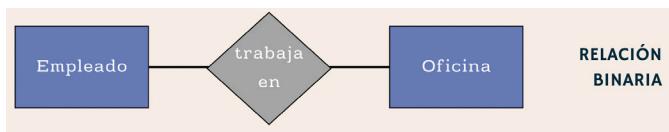
Los conjuntos de relaciones que involucran dos conjuntos de entidades se llaman relaciones binarias (o de grado dos). La mayoría de las relaciones en una base de datos es de este tipo.

Los conjuntos de relaciones pueden involucrar a más de dos conjuntos de entidades.

Nombre de la relación	Grado
Unarias o Unitarias o de Reflexión	1
Binarias	2
Ternarias	3
N -arias	N

Algunos ejemplos serían:



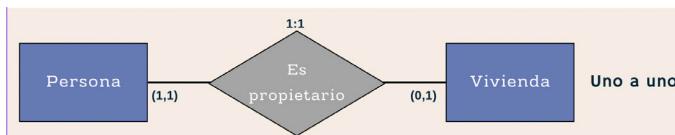


b) Cardinalidad

Cuando un conjunto de relaciones involucrados o más grupos de entidades, la cardinalidad de la correspondencia indica cuántas entidades pueden estar conectadas a una entidad en particular. En el caso de un conjunto de relaciones binarias y los grupos de entidades A y B, las cardinalidades pueden ser:

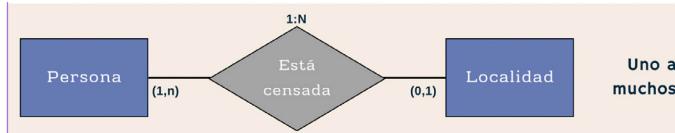
Uno a Uno: (1:1)

Un registro de una entidad A se relaciona con solo un registro en una entidad B.



Uno a Varios: (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.

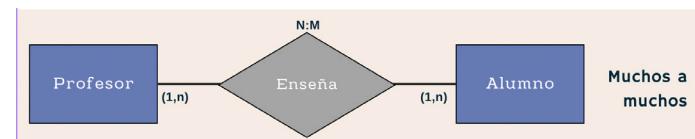


Varios a uno: (N:1)

Una entidad en A se relaciona exclusivamente con una entidad en B. Pero una entidad en B se puede relacionar con 0 o muchas entidades en A.

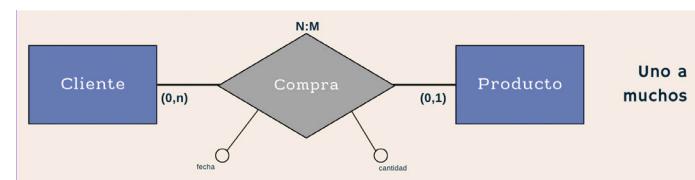
Varios a varios: (N:M)

Una entidad en A se puede relacionar con 0 o con muchas entidades en B y viceversa.



c) Atributos propios de una relación

También las relaciones pueden tener atributos. Son aquellos atributos cuyo valor sólo se puede obtener en la relación, puesto que dependen de todas las entidades que participan en la relación.



Entidades fuertes y entidades débiles

Una entidad fuerte es aquella que no necesita de otra entidad débil para existir.

Una entidad débil es aquella que sí que necesita de otra para existir

Las entidades fuertes se representarán con un rectángulo y las débiles con un rectángulo de línea doble, por ejemplo:

Diagrama de entidades fuertes y débiles:

```

    graph LR
        A[PRÉSTAMO] --- D{tiene}
        D --- B[PAGO]
    
```

PRÉSTAMO

PAGO

PAGO			PRÉSTAMO	
Nro	Importe	Fecha	Nro	Total
1	\$1200	02/02/2020	1	\$100.000
2	\$2500	02/02/2020	2	\$125.000
1	\$1500	02/03/2020	3	\$230.000
1	\$1000	03/03/2020		

Supongamos el conjunto de entidades Pago, que vemos en la imagen anterior, con los atributos Nro, Importe y Fecha. Se trata de almacenar la información de los distintos pagos necesarios para devolver un préstamo.

Cada pago, es un pago distinto al resto de pagos. Pero en estas condiciones existen números de pagos repetidos, incluso el importe y la fecha podrían repetirse, como vemos en el ejemplo.

Los pagos son diferentes y corresponden a préstamos diferentes.

¿Cómo podemos identificar a qué préstamo corresponde un pago? Sencillamente no podemos, puesto que no hay una clave primaria que identifique cada pago. Nos encontramos ante un ejemplo de entidad débil, ya que cada pago depende de la existencia de un préstamo para tener significado. Una posible solución podría ser agregar una clave primaria a la entidad PAGOS que permita identificar cada pago, como se ve a continuación.

PAGO			
*ID	Nro	Importe	Fecha
23	1	\$1200	02/02/2020
24	2	\$2500	02/02/2020
25	1	\$1500	02/03/2020
26	1	\$1000	03/03/2929

En este caso, si podemos diferenciar cada pago del resto de los pagos.

Sin embargo, seguimos sin poder darle significado, pues no se puede relacionar el pago con el préstamo.

Por ende, Pago es una entidad débil en existencia y en identificación

La razón es simple: existen pagos porque necesariamente, hay un préstamo. ¿Por qué debería pagar si no me han prestado?

Pero entonces, ¿cómo podríamos dar significado a registros de una entidad débil y relacionarlos a una entidad fuerte?

PAGO			
Prestamo	Pago	Importe	Fecha
1	1	\$1200	02/02/2020
2	2	\$2500	02/02/2020
3	1	\$1500	02/03/2020
4	1	\$1000	03/03/2929
1	2	\$1200	02/03/2020
...

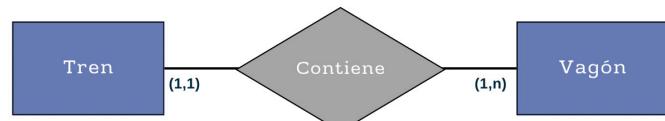
La clave primaria de la entidad débil se formará con la clave primaria de la entidad fuerte más el atributo que identifica a los datos de la entidad débil.

Relaciones de Dependencia

Las relaciones de dependencia son las que se producen entre las entidades fuertes y las débiles.

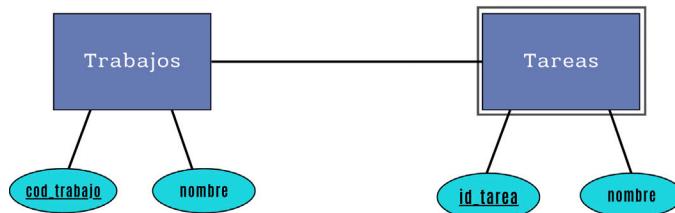
Existen dos tipos, y algo ya te hemos adelantado con el ejemplo anterior:

De existencia: La entidad débil no puede existir sin la fuerte de la que depende.

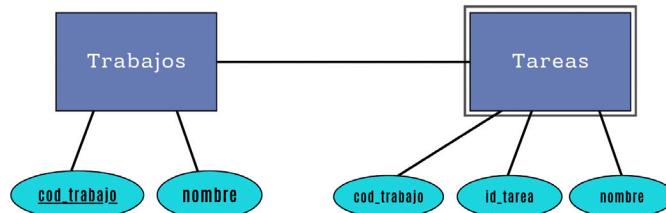


El vagón por sí solo puede trabajar, pero su existencia depende del tren.

De identificación: Cuando además de por existencia, la entidad débil no es capaz de identificarse por sí misma y depende del identificador de la entidad fuerte.



Para identificar la tarea sumamos la clave primaria del Trabajo.



La Cardinalidad de una entidad débil siempre es la siguiente:

- **(1,1).** En el lado de la entidad fuerte, haciendo referencia a que cada elemento de la entidad débil se relaciona con uno y solamente uno de los ejemplares de la entidad fuerte.
- **(0,n) o (1,n).** En el lado de la entidad fuerte. Lo habitual es (0,n) indicando que los ejemplares de la entidad fuerte no tienen por qué relacionarse con ninguno de los ejemplares de la entidad débil.

Para comprender mejor la Cardinalidad, te dejaremos algunos ejemplos prácticos (¿te animas a hacer los MER de estos ejemplos?):

Uno a Uno (1:1):

Un empleado tiene un único número de seguro social asignado.

Un cliente tiene una única dirección de envío.

Uno a Muchos (1:N):

Un autor puede escribir varios libros.

Un departamento tiene muchos empleados.

Muchos a Muchos (N:N):

Varios estudiantes pueden inscribirse en varios cursos.

Varios productos pueden ser parte de varias órdenes de compra.

En los ejemplos anteriores, la cardinalidad se expresa en términos de relaciones entre entidades. Por ejemplo, en el caso del autor y el libro, un autor puede tener una relación de uno a muchos con los libros, lo que significa que un autor puede escribir varios libros (1:N). En el caso de los estudiantes y los cursos, hay una relación de muchos a muchos, lo que significa que varios estudiantes pueden inscribirse en varios cursos (N:N).

Es importante tener en cuenta que la cardinalidad puede variar según el contexto y los requisitos específicos del sistema de información o de la base de datos en particular. Es esencial determinar la cardinalidad correcta para establecer relaciones adecuadas entre las entidades y garantizar un modelo de datos preciso y eficiente.

DEL MODELO CONCEPTUAL AL MODELO LÓGICO

Luego de haber realizado el análisis de la situación real, modelando la misma mediante un Esquema Conceptual, a través del MER, nos abocaremos a la obtención del modelo lógico de nuestra Base de Datos.

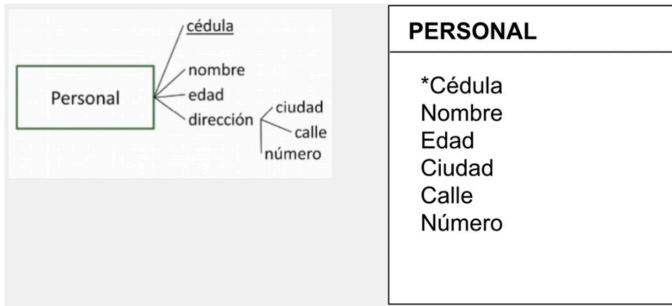
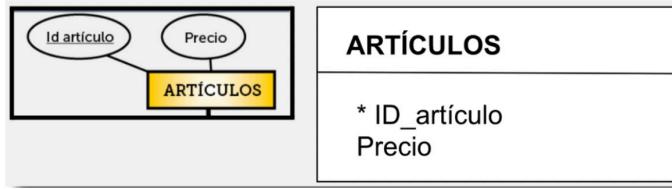


En este apunte te contaremos los pasos a seguir detalladamente a partir de diversos ejemplos.

ENTIDADES

Todas las entidades regulares presentes en el MER se transforman en tablas en el modelo lógico, manteniendo el número y tipo de los atributos, así como las claves primarias.

- El nombre de la tabla será el nombre de la entidad.
- Cada atributo de la entidad será un campo de la tabla.
- Los atributos que definen la clave primaria serán, también, la clave primaria de la tabla.
- En el caso que hayan atributos compuestos se descomponen el atributo en campos. (lo vamos a ver mejor en el ejemplo de Personal)

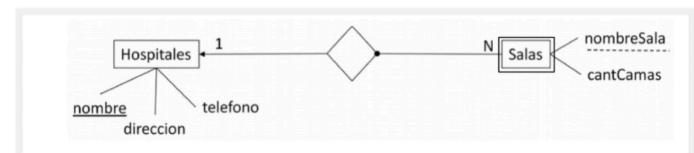


Las **entidades débiles** también se convierten en tablas en el modelo relacional, manteniendo el **número y tipo de los atributos**, pero su clave primaria se forma por la composición de su clave primaria con la clave primaria de la entidad regular de la cual depende.

requisitos específicos del sistema de información o de la base de datos en particular. Es esencial determinar la cardinalidad correcta para establecer relaciones adecuadas entre las entidades y garantizar un modelo de datos preciso y eficiente.

DEL MODELO CONCEPTUAL AL MODELO LÓGICO

Luego de haber realizado el análisis de la situación real, modelando la misma mediante un Esquema Conceptual, a través del MER, nos abocaremos a la obtención del modelo lógico de nuestra Base de Datos.



HOSPITAL	SALAS
<ul style="list-style-type: none"> *NombreHospital Dirección Telefono 	<ul style="list-style-type: none"> *NombreHospital *NombreSala Dirección Telefono

RELACIONES

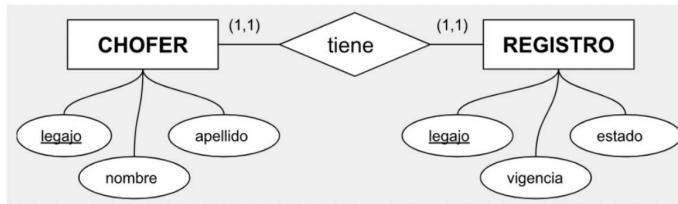
Para convertir las relaciones a tablas, hay que tener presente su cardinalidad: 1:1, 1-N, N:N

Dependiendo de cómo se definan las cardinalidades de las relaciones, éstas pueden generar una tabla o por el contrario traducirse en campos dentro de las tablas asociadas a las entidades que participan en dicha relación.

RELACIONES 1:1

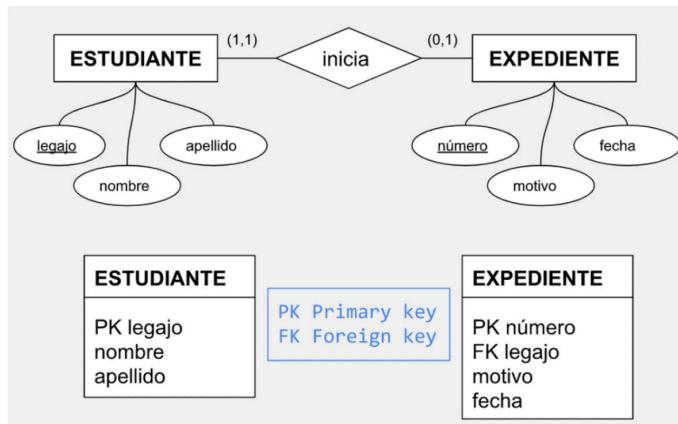
En este tipo de relaciones las entidades podrían utilizar la misma clave o cada cual la suya.

Si las dos entidades tienen el mismo identificador, entonces se transforman en una única tabla por la agregación de los atributos de las dos entidades y la clave es la clave de las entidades (es la misma en ambas).



CHOFER
*legajo nombre apellido vigencia estado

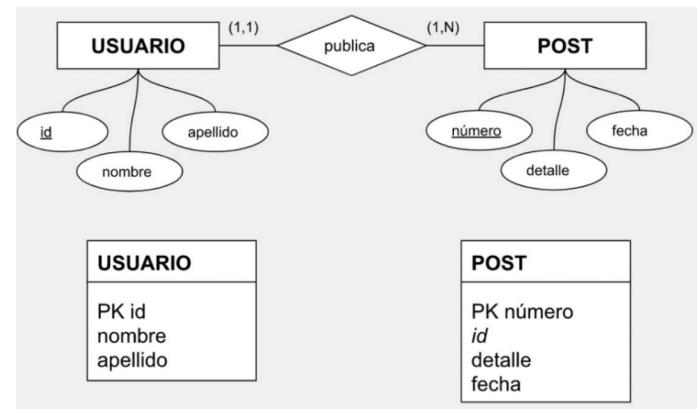
Si las dos entidades tienen distinto identificador, entonces cada entidad se transforma en una tabla con clave principal y cada tabla tendrá como clave foránea el identificador de la otra tabla con la cual está relacionada.



RELACIONES 1:N

En el tipo de relación binaria 1:N, no es necesario generar otra tabla, lo que se hace es incluir en la entidad del lado de N el identificador de la otra tabla y los posibles campos de la relación si los hubiere.

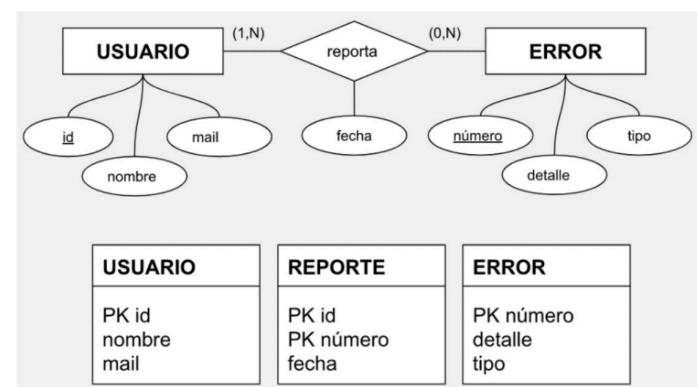
Veamos cómo funciona:



RELACIONES N:N

En este caso crearemos una nueva tabla para representar la relación, agregando la clave primaria de cada una de las tablas, además de los campos que correspondan a atributos de la relación.

La clave primaria de la nueva tabla será la combinación de las claves primarias de las tablas que participan en la relación.



RELACIONES DE HERENCIA

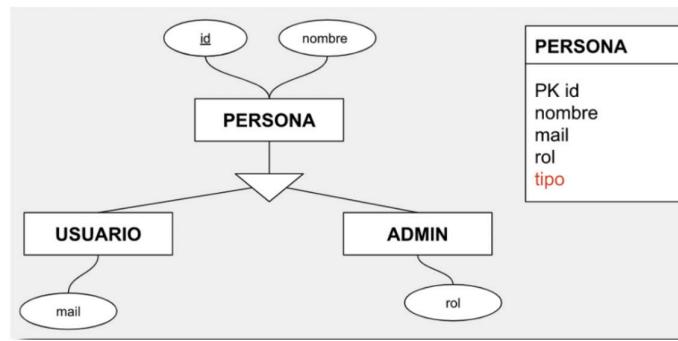
En el caso de representaciones de Herencia normalmente se generan las siguientes tablas: una para la super entidad (nivel superior), con los atributos comunes, y otra

tabla para cada una de las entidades subordinadas, con la misma clave que la entidad superior y los atributos propios de esa sub-entidad. Por ejemplo:

PERSONA	USUARIO	ADMIN
PK id nombre	FK id mail	FK id rol

Con esta solución un Usuario nunca puede ser Admin, ni viceversa.... qué opción hay para poder representar que es posible que un Usuario sea Admin?

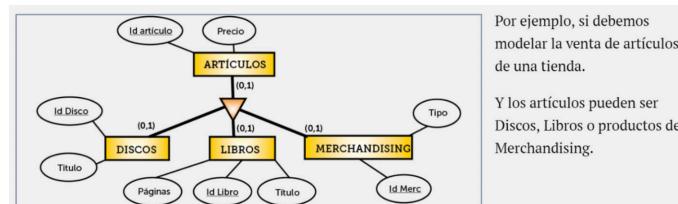
Para solucionarlo, tenemos la opción de crear una tabla con la super entidad, sus atributos y claves e incorporar los atributos de las entidades subordinadas. Agregando además el campo tipo que permitirá distinguir si la persona es USUARIO o ADMIN.



ESPECIALIZACIÓN

La relación de Especialización implica la designación de subgrupos dentro de un conjunto de entidades.

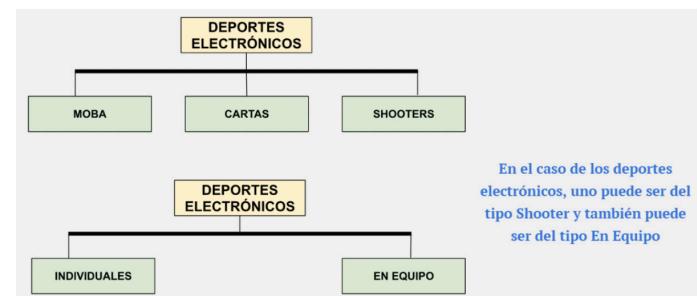
Puede incluir subgrupos de entidades que se diferencian de alguna forma de las otras entidades del conjunto. Se produce un Diseño descendente: refinamiento a partir de un conjunto de entidades inicial en sucesivos niveles.



Por ejemplo, si debemos modelar la venta de artículos de una tienda.

Y los artículos pueden ser Discos, Libros o productos de Merchandising.

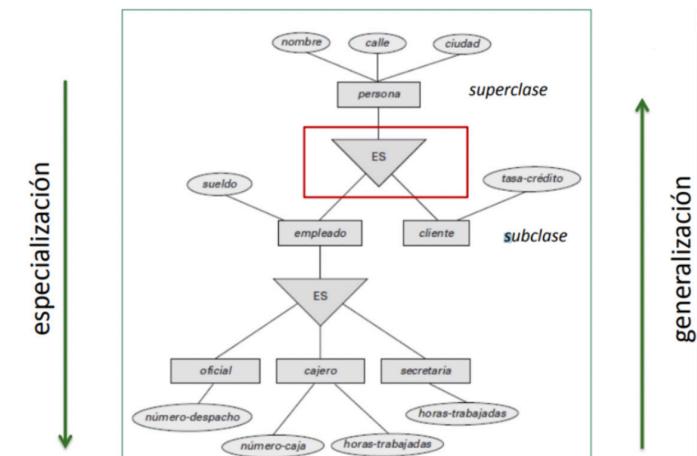
Se puede aplicar repetidamente la especialización para refinar el esquema de diseño. Por ejemplo:



GENERALIZACIÓN

La generalización es una inversión simple de la especialización.

Gráficamente se produce un Diseño ascendente: varios conjuntos de entidades se sintetizan en un conjunto de entidades de nivel más alto basado en características comunes.



HERENCIA DE ATRIBUTOS

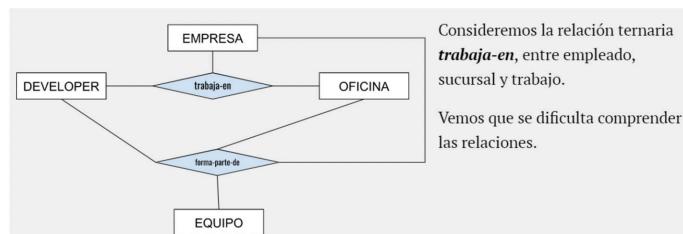
Los atributos de los conjuntos de entidades de nivel más alto son heredados por los conjuntos de entidades de nivel más bajo.

Por ejemplo: cliente y empleado heredan los atributos de la entidad persona.

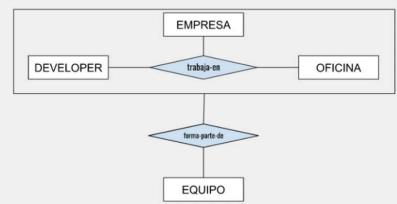
AGREGACIÓN

Una limitación del MER es que no resulta posible expresar relaciones entre relaciones.

Para resolverlo se utiliza la Agregación, una abstracción a través de la cual las relaciones se tratan como entidades de nivel más alto.



Usando la agregación, podríamos agrupar las entidades de la relación **trabaja-en** y obtenemos la siguiente representación



NOTACIÓN DE CHEN

Peter Chen (también conocido como Peter Pin-Shan Chen) actualmente se desempeña como miembro de la facultad de la Universidad Carnegie Mellon ubicada en Pittsburgh y se le atribuye el desarrollo del modelo ER para el diseño de bases de datos en los 70. Mientras trabajaba como profesor adjunto en la Escuela de Administración y Dirección de Empresas Sloan del MIT, publicó un documento influyente en 1976 llamado "Modelo entidad-relación: hacia una visión unificada de los datos".

GLOSARIO DE SÍMBOLOS

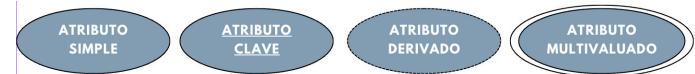
Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este. Piensa en las entidades como si fueran sustantivos. Por ejemplo: un cliente, estudiante, auto o producto.

Entidad

Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este. Piensa en las entidades como si fueran sustantivos. Por ejemplo: un cliente, estudiante, auto o producto.

Atributo

Una propiedad o característica de una entidad.



Categorías de los atributos: los atributos se clasifican en simples, compuestos y derivados, así como de valor único o de valores múltiples.

- **Simples:** significa que el valor del atributo es mínimo y ya no puede dividirse, como un número de teléfono.
- **Compuestos:** los subatributos surgen de un atributo.
- **Derivados:** los atributos se calculan o derivan de otro atributo, por ejemplo, la edad se calcula a partir de la fecha de nacimiento.
- **Valores múltiples:** se denota más de un valor del atributo, como varios números de teléfono para una persona
- **Valor único:** contienen solo un valor de atributo. Los tipos se pueden combinar, por ejemplo, puede haber atributos de valor único simples o atributos de múltiples valores compuestos.

RELACIONES

Asociaciones que permiten vincular entidades entre sí. Usar verbos para relaciones y representarlas con rombos.



CARDINALIDAD

Define los atributos numéricos de la relación entre dos entidades o conjuntos de entidades. Las tres relaciones cardinales principales son uno a uno, uno a muchos y muchos a muchos.

Un ejemplo de uno a uno sería un estudiante asociado a una dirección de correo electrónico. Un ejemplo de uno a muchos (o muchos a uno, en función de la dirección de la relación) sería un estudiante que se inscribe en muchos cursos, y todos esos cursos se asocian a ese estudiante en particular. Un ejemplo de muchos a muchos sería los estudiantes en grupo están asociados a múltiples miembros de la facultad y a su vez los miembros de la facultad están asociados a múltiples estudiantes.

Un último ejemplo de representación de un Modelo Entidad-Relación con Notación de Chen

A continuación, se presenta un ejemplo práctico para diseñar una base de datos que apoye la gestión de un sistema de ventas. Este trabajo fue solicitado por una empresa que necesitaba llevar un seguimiento de proveedores, clientes, productos y ventas. A continuación, se detallan las características de cada una de las entidades:

Un proveedor posee un número de identificación tributaria (RUT), nombre, dirección, teléfono y página web.

Un cliente también tiene un RUT, nombre, dirección, y uno o más teléfonos de contacto.

La dirección se compone de calle, número, comuna y ciudad.

Un producto se identifica por un id único, nombre, precio actual, stock y el nombre del proveedor que lo suministra. Además, cada producto pertenece a una sola categoría.

Una categoría se identifica por un id, nombre y descripción, la cantidad vendida y el monto total por el producto.

Para fines de contabilidad, se debe registrar la información de cada venta con un id, fecha, cliente, descuento y monto final. Además, se debe guardar el precio al momento de la ven-

