ISTRED
INSTITUTO SUPERIOR TÉCNICO REY DAVID

PERIODO II
2023 - 2024

GUÍA DE APRENDIZAJE DEL ESTUDIANTE
BASE DE DATOS

ING. FERNANDO DOMÍNGUEZ RAMOS

GESTIÓN DE BASE DE DATOS
2DO. SEMESTRE

Contenido

I. DATOS GENERALES DE LA ASIGNATURA 7

Información General 7

Requisitos 7

Descripción de la asignatura. 7

Objetivos de la asignatura: 8

Introducción de la Asignatura 8

I. DATOS GENERALES DE LA UNIDAD 1 10

Información General 10

Resultado de Aprendizaje 10

Metodología 10

DESARROLLO 11

Unidad 1: Introducción a sistemas de bases de datos 11

Tema 1: Introducción a los conceptos de base de datos 11

Definición de una Base de Datos 11

Funciones y Beneficios de una Base de Datos 12

Historia de las Bases de Datos 12

Orígenes y Precursores 13

Evolución y Adopción Generalizada 13

Tendencias Actuales y Futuras 14

Normalización de Bases de Datos 15

Sistemas de Gestión de Bases de Datos (DBMS) 16

Selección de un DBMS 17

Bases de Datos Relacionales 18
Principios Fundamentales de las Bases de Datos Relacionales 18
SQL (Structured Query Language) 19
Funciones Clave de SQL 19
Importancia y Uso de SQL 20
Características Principales de las Base de Datos 21
ACTIVIDADES PRÁCTICAS 22
FORO DE DEBATE 23
PRÁCTICA INVESTIGATIVA 23
PRÁCTICA TEST 24
Tema 2: Modelado de Datos Lógicos de Base de datos 25
Importancia del Modelado de Datos Lógicos en el Desarrollo de Software 26
Conceptualizaciones: 26
Modelo Conceptual de Base de Datos. 29
Transición a los Modelos Lógico y Físico 31
Modelo Entidad-Relación (ER) 31
Elementos Claves del Modelo ER con Ejemplos 32
ACTIVIDADES PRÁCTICAS 34
FORO DE DEBATE 35
PRÁCTICA INVESTIGATIVA 35
PRÁCTICA TEST 36
II. DATOS GENERALES DE LA UNIDAD 2 37
Información General 37

Resultado de Aprendizaje 37

Metodología 37

DESARROLLO 38

Unidad 2: Aplicación de Modelos de Base de Datos 38

Tema 1: Modelo Entidad Relación 38

Elementos Claves del Modelo ER con Ejemplos 38

Transición del Modelo ER al Diseño Lógico y Físico 40

Historia 40

Importancia del Modelado Entidad Relación 42

Cardinalidades en los Modelos Entidad Relación 44

Modelado de datos físicos de base de datos 46

Introducción Instalación del sistema de base de datos 49

ACTIVIDADES PRÁCTICAS 52

FORO DE DEBATE 53

PRÁCTICA INVESTIGATIVA 53

PRÁCTICA TEST 54

Tema 2: Sistemas de Gestión de Base de Datos 55

Definiciones y Componentes de un SGBD 55

Configuración de SGBD 5

ACTIVIDADES PRÁCTICAS 11

FORO DE DEBATE 12

PRÁCTICA INVESTIGATIVA 12

PRÁCTICA TEST 13

III. DATOS GENERALES DE LA UNIDAD 3 14
Información General 14
Resultado de Aprendizaje 14
Metodología 14
DESARROLLO 15
Unidad 3: Estructuras internas de almacenamiento y estructura de índices 15
Tema 1: Arquitectura del sistema de base de datos 15
Instancia de Sistemas de Base de Datos 15
ACTIVIDADES PRÁCTICAS 22
FORO DE DEBATE 23
PRÁCTICA INVESTIGATIVA 23
PRÁCTICA TEST 24
Tema 2: Objetos en Bases de Datos 25
Principales Objetos de una Base de Datos: 25
Procedimientos Almacenados: Los procedimientos almacenados son conjuntos de instrucciones SQL que se almacenan y pueden ser ejecutados repetidamente 26
Procedimientos almacenados en una base de datos 27
Ejemplo Avanzado: 28
Funciones en una Base de datos 29
Tipos de Funciones: 29
Disparadores y Paquetes en una Base de Datos 31
Ejemplo de Disparador: 31

ACTIVIDADES PRÁCTICAS 34
FORO DE DEBATE 35
PRÁCTICA INVESTIGATIVA 35
PRÁCTICA TEST 36
IV. DATOS GENERALES DE LA UNIDAD 4 37
Información General 37
Resultado de Aprendizaje 37
Metodología 37
DESARROLLO 38
Unidad 4: Casos prácticos 38
Tema 1: Práctica de Comandos DML y DDL 38
Desarrollo de casos prácticos. DML y DDL 38
Ejemplo DDL: 39
Ejemplo DML: 39
Metodología para Desarrollar Casos Prácticos: 40
Transacciones y Técnicas de Recuperación 40
Ejemplo de Transacción: 41
Procesamiento de transacciones, COMMIT, ROLLBACK. 42
Ejemplo de COMMIT: 42
ACTIVIDADES PRÁCTICAS 44
FORO DE DEBATE 45
PRÁCTICA INVESTIGATIVA 45
PRÁCTICA TEST 46

Tema 2: Introducción a Seguridad en Base de Datos 47
Componentes Clave de la Seguridad en Bases de Datos 47
Importancia de la Seguridad en Bases de Datos 48
Implementación de Scripts, aplicación mediante SQL. 49
Componentes Clave de la Implementación de Scripts 49
Accesos autorizados: Roles y Privilegios 50
Implementación de Scripts de seguridad y autorización 52
Optimización de consultas 54
ACTIVIDADES PRÁCTICAS 56
FORO DE DEBATE 57
PRÁCTICA INVESTIGATIVA 57
PRÁCTICA TEST 58
V. Bibliografía 59
Bibliografía básica 59
Bibliografía de consulta 59
Anexo 1: Modelo de Foro 60
Anexo 2: Esquema de Ensayo 61
Anexo 3: Esquema de Taller/Investigación 62
Anexo 4: Rubrica de taller práctico 63
Anexo 5: Rúbrica de Foro 64
Anexo 6: Rubrica de Ensayo 65
Anexo 7: Rúbrica de Taller/Investigación 66

I. DATOS GENERALES DE LA ASIGNATURA

Información General

1.1.	Asignatura	Base de Datos					
1.2.	Carrera	Gestión de Base de Datos					
1.3.	Código de asignatura	GBD2239					
1.4.	Créditos	1,2					
1.5.	Nivel	Segundo					
1.6.	Detalle de horas	ACD	24	AA	16	APE	20

Requisitos

Prerrequisitos		Correquisitos	
Asignatura	Código	Asignatura	Código
Teoría de Sistemas	GBD1236	N/A	N/A

Descripción de la asignatura.

La asignatura "Base de Datos" es un componente fundamental en la formación de estudiantes de educación superior a nivel técnico. En esta disciplina, los estudiantes adquieren conocimientos esenciales para la organización, gestión y manipulación eficiente de datos en contextos informáticos. La base de datos es un pilar en la ciencia de la computación y la tecnología de la información, ya que es fundamental para el almacenamiento y recuperación de información en una amplia gama de aplicaciones, desde sistemas empresariales hasta aplicaciones web y móviles.

Durante el curso, los estudiantes explorarán conceptos clave relacionados con bases de datos, como la estructura de datos, la normalización, la gestión de transacciones y la optimización de consultas. También se familiarizarán con lenguajes de consulta, como SQL (Structured Query Language), que les permitirán diseñar y consultar bases de datos de manera efectiva. Además, aprenderán sobre sistemas de gestión de bases de datos (DBMS), que son herramientas esenciales en el mundo real para administrar bases de datos a gran escala. Esta asignatura preparará a los estudiantes para desempeñar un papel crucial en el desarrollo y mantenimiento de sistemas de información eficaces y les proporcionará habilidades valiosas en el mercado laboral de la tecnología.

En resumen, la asignatura "Base de Datos" es un componente esencial en la formación de estudiantes de educación superior de nivel técnico, ya que proporciona una base sólida en la gestión de datos y el diseño de bases de datos. Los conocimientos adquiridos en este curso son aplicables en diversos campos, y los estudiantes estarán preparados para abordar desafíos reales en el mundo de la tecnología de la información, contribuyendo al desarrollo y la eficiencia de sistemas informáticos en una variedad de entornos.

Objetivos de la asignatura:

Introducir al alumno en el conocimiento general en la creación de aplicaciones que den soporte a distintos procesos de negocio utilizando el lenguaje Transact/SQL avanzado en una base de datos Mysql o Microsoft SQL Server.

Recuperar, insertar, actualizar y eliminar información de una base de datos utilizando Transact/SQL y aplicando múltiples condiciones de comparación o funciones para el manejo de campos tipo fecha. construir programas estructurados y manejar los errores utilizando el lenguaje Transact-SQL

Crear cursores con la capacidad de procesar grandes volúmenes de transacciones con el mejor rendimiento posible. Implementar instrucciones Transact/SQL y de programación mediante procedimientos almacenados, funciones y desencadenadores para optimizar las operaciones en la base de datos y las operaciones DML.

Aplicar la teoría en la práctica: Los estudiantes estarán preparados para aplicar sus conocimientos en situaciones reales. Aprenderán a diseñar y desarrollar bases de datos desde cero, así como a administrar y mantener bases de datos existentes. También estarán equipados para evaluar y seleccionar sistemas de gestión de bases de datos adecuados para diferentes aplicaciones.

Fomentar la resolución de problemas y el pensamiento crítico: La asignatura promoverá la resolución de problemas a través del diseño y la implementación de bases de datos, lo que permitirá a los estudiantes desarrollar habilidades de pensamiento crítico y la capacidad de abordar desafíos en el mundo real relacionados con la gestión de datos y sistemas de información.

Introducción de la Asignatura

En la era digital actual, la gestión eficiente de la información se ha convertido en un activo crítico para organizaciones de todos los sectores. La asignatura "Base de

Datos" se erige como un pilar fundamental en la formación de estudiantes de educación superior a nivel técnico, brindando los conocimientos y habilidades necesarios para abordar los desafíos de almacenamiento, organización y acceso a datos en el mundo de la informática. La base de datos es el núcleo que sustenta la mayoría de las aplicaciones y sistemas informáticos que utilizamos a diario, desde sistemas de gestión empresarial hasta aplicaciones web y móviles.

En este curso, los estudiantes se embarcarán en un viaje de descubrimiento que los llevará desde los conceptos básicos de las bases de datos hasta la maestría en el diseño de sistemas de almacenamiento de datos eficientes. Explorarán conceptos cruciales, como la normalización, la integridad de los datos y los sistemas de gestión de bases de datos, y dominarán el lenguaje SQL para la creación y manipulación de datos. Nuestro objetivo es equipar a los estudiantes con las habilidades necesarias para convertirse en profesionales capaces de diseñar y gestionar bases de datos que impulsen la innovación y la eficiencia en una amplia variedad de aplicaciones tecnológicas. Esta asignatura se enfoca no solo en la teoría, sino también en la aplicación práctica, preparando a los estudiantes para enfrentar los desafíos del mundo real en el apasionante campo de la gestión de datos.

I. DATOS GENERALES DE LA UNIDAD 1

Información General

1.1.	Nivel	Segundo					
1.2.	Nombre Unidad	Introducción a sistemas de bases de datos.					
1.3.	Tema 1 – Unidad	Introducción a los conceptos de base de datos.					
1.4.	Tema 2 – Unidad	Introducción al diseño de base de datos.					
1.5.	Unidad organizacional	Unidad Profesional					
1.6.	Unidad	1					
1.7.	Total, Horas Unidad	15					
1.8.	Detalle de horas Unidad	ACD	6	AA	4	APE	5

Resultado de Aprendizaje

Al finalizar esta unidad, los estudiantes serán capaces de recuperar, insertar, actualizar y eliminar información de una base de datos utilizando Transact/SQL y aplicando múltiples condiciones de comparación o funciones para el manejo de campos tipo fecha. construir programas estructurados y manejar los errores utilizando el lenguaje Transact-SQL.

Metodología

La Metodología de se basa en un enfoque activo y colaborativo. Incluye sesiones de clase estructuradas en torno a temas específicos, lecturas y tareas, estudio de casos, proyectos prácticos y debates en clase. Los estudiantes también se involucran en trabajos en grupo y mantienen diarios de reflexión. Se fomenta la participación en debates y preguntas y respuestas, y se proporcionan recursos en línea para una comprensión más profunda. La evaluación se realiza mediante pruebas, proyectos y una evaluación final, con retroalimentación continua para guiar el aprendizaje. Esta metodología busca que los estudiantes comprendan los conceptos fundamentales y puedan aplicar eficazmente las metodologías de desarrollo de software en situaciones prácticas.

DESARROLLO

Unidad 1: Introducción a sistemas de bases de datos

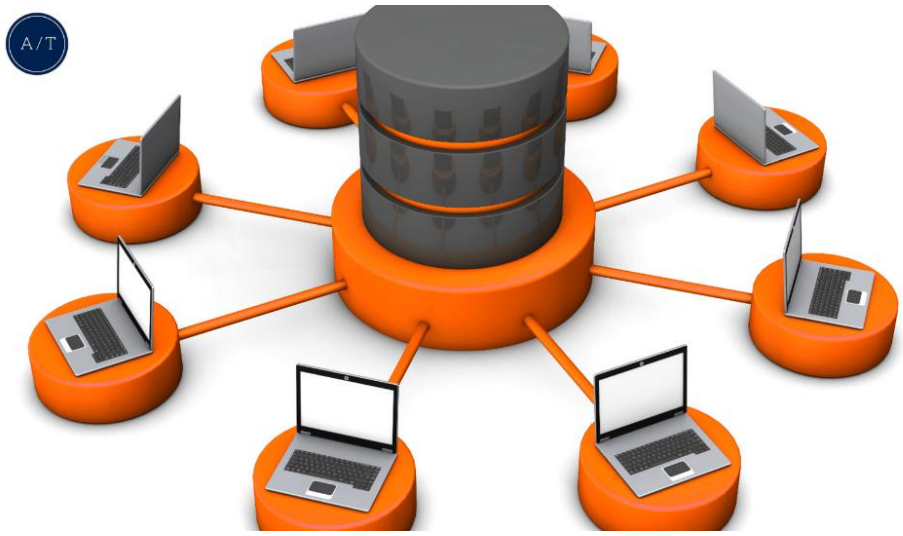
Tema 1: Introducción a los conceptos de base de datos

En el mundo de la tecnología de la información, las bases de datos son una parte esencial de la gestión de datos. Una base de datos es una colección organizada de información que permite almacenar, acceder y gestionar datos de manera eficiente en aplicaciones informáticas.

Definición de una Base de Datos

Una base de datos consta de tablas que almacenan información relacionada. Cada tabla contiene registros, que representan unidades de datos, y campos que describen atributos específicos de esos registros.

Imagen 1. Base de Datos.



Fuente: https://atlantictech.io/base-de-datos-como-se-dividen/

En el contexto de la gestión de información, una base de datos es un componente esencial que proporciona un medio organizado y eficiente para almacenar, gestionar y recuperar datos. Imagina la base de datos como un almacén digital que alberga información estructurada, permitiendo a usuarios y aplicaciones acceder a los datos de manera rápida y efectiva. Su importancia radica en su capacidad para ofrecer una estructura coherente que facilita la organización y recuperación de datos, garantizando la integridad y consistencia de la información.

Funciones y Beneficios de una Base de Datos

Las funciones de una base de datos van más allá del simple almacenamiento de datos; incluyen la capacidad de realizar consultas complejas, actualizaciones eficientes y mantener la integridad de la información. La normalización, un proceso clave, asegura que la base de datos esté organizada de manera eficiente, reduciendo redundancias y mejorando la calidad de los datos. En conjunto, estos elementos y funciones hacen de la base de datos una herramienta esencial en el mundo de la informática, respaldando la toma de decisiones informada y eficaz en una variedad de contextos.

Imagen 2: Ventajas y Desventajas.



Fuente: https://francoise1098.blogspot.com/2014/07/ventaja-y-desventajas-de-las-bases-de.html

Historia de las Bases de Datos

El concepto de bases de datos se remonta a la década de 1960, con el trabajo pionero de Edgar F. Codd en el modelo relacional, que sienta las bases para la mayoría de los sistemas de bases de datos utilizados hoy en día.

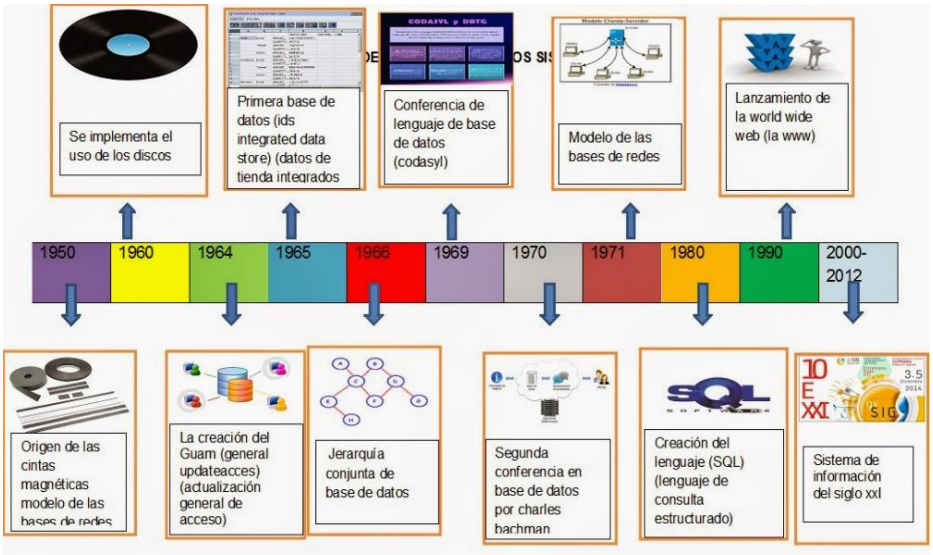
Orígenes y Precursores

La historia de las bases de datos se remonta a los primeros intentos de organizar y gestionar datos de manera estructurada. En las décadas de 1960 y 1970, pioneros como Charles Bachman y Edgar F. Codd sentaron las bases para lo que hoy conocemos como sistemas de gestión de bases de datos (DBMS). Bachman desarrolló el Sistema Integrado de Datos (IDS), un precursor de los modernos DBMS, mientras que Codd introdujo el modelo relacional en 1970, un hito fundamental que revolucionó la gestión de datos.

Evolución y Adopción Generalizada

A medida que la tecnología avanzaba, la necesidad de gestionar grandes volúmenes de datos se volvía cada vez más apremiante. En la década de 1980, el modelo relacional ganó popularidad y se convirtió en el estándar de facto para las bases de datos. Con el tiempo, surgieron varios DBMS comerciales, como Oracle, Microsoft SQL Server y MySQL, que permitieron a las organizaciones gestionar datos de manera más eficiente. La década de 1990 vio la adopción generalizada de bases de datos en diversas aplicaciones, desde sistemas empresariales hasta la explosión de la web.

Imagen 3: Evolución de Base de Datos.



Fuente: https://bdsena893311.blogspot.com/2015/02/base-de-datos-taller-1_9.html

Tendencias Actuales y Futuras

Hoy en día, vivimos en una era de diversificación de bases de datos. La llegada de NoSQL introdujo modelos de datos flexibles y escalables, rompiendo con la rigidez del modelo relacional. Además, la nube ha transformado la forma en que almacenamos y accedemos a datos, permitiendo una mayor flexibilidad y escalabilidad. A medida que avanzamos, la inteligencia artificial y el análisis predictivo están influyendo en el diseño y la gestión de bases de datos, señalando un futuro emocionante y dinámico para esta disciplina esencial en el mundo de la informática.

La adopción generalizada de la nube ha transformado la gestión de bases de datos. Empresas y organizaciones migran a entornos en la nube para aprovechar la flexibilidad, escalabilidad y accesibilidad que ofrecen. Plataformas como Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform han desempeñado un papel crucial al proporcionar servicios de bases de datos gestionados que simplifican la administración y permiten a las empresas enfocarse en la innovación y no en la infraestructura.

Imagen 4: Características de Amazon



Fuente: https://explodat.cl/Analytics/cloud-computing/que-es-aws-y-para-que-sirve-introduccion-de-la-a-a-la-z/

Otra tendencia emocionante es la creciente integración de inteligencia artificial (IA) y analítica predictiva en la gestión de bases de datos. Las capacidades avanzadas de análisis de datos permiten descubrir patrones, prever tendencias y tomar decisiones más informadas. Los sistemas de bases de datos están evolucionando para soportar

análisis en tiempo real y aprendizaje automático, lo que promete transformar la forma en que interactuamos y extraemos valor de nuestros datos.

Imagen 5. Uso de Inteligencia Artificial.



Fuente: www.europarl.europa.eu/news/es/headlines/society/20200827STO85804/

Normalización de Bases de Datos

La normalización es un proceso de diseño de bases de datos que tiene como objetivo eliminar la redundancia de datos y garantizar la integridad de la información. Se logra mediante la organización eficiente de tablas y relaciones.

Imagen 6. Ejemplo de Normalización en Base de datos.



Fuente: https://bookdown.org/paranedagarcia/database/normalizacion.html

La normalización de bases de datos es un proceso esencial que tiene como objetivo organizar la información de manera eficiente y reducir la redundancia en las bases de datos. Introducida por Edgar F. Codd en la década de 1970, la normalización busca

garantizar la integridad de los datos y mejorar la eficiencia de las operaciones de consulta y actualización. Este proceso se basa en principios que eliminan la redundancia al dividir las tablas en estructuras más pequeñas y relacionadas, minimizando así la duplicación de datos y asegurando una representación coherente de la realidad en la base de datos.

La normalización sigue una serie de formas normales, cada una diseñada para abordar aspectos específicos de la redundancia en los datos. Las formas normales, desde la primera (1NF) hasta la quinta (5NF), definen criterios específicos para la organización de datos. El proceso de normalización implica identificar dependencias funcionales, eliminar duplicados y asegurar que cada atributo sea completamente dependiente de la clave primaria. A medida que se avanza en las formas normales, se mejora la estructura de la base de datos, reduciendo el riesgo de anomalías y asegurando la consistencia de los datos.

Imagen 7: Problema de redundancia en Base de Datos

matricula	nombre	dirección	telefono	curso	código	carrera
100	juan	ongolmo 340, concepción	78872890	base de datos	bd1	ingeniería civil informatica
100	juan	ongolmo 340, concepción	78872890	estadística	e2	ingeniería civil informatica
100	juan	ongolmo 340, concepción	78872890	analitica	a5	ingeniería civil informatica
200	ana	san martín 840, santiago	78342367	estadística	e2	ingeniería comercial
200	ana	san martín 840, santiago	78342367	analitica	a5	ingeniería comercial

Fuente: https://bookdown.org/paranedagarcia/database/normalizacion.html

Sistemas de Gestión de Bases de Datos (DBMS)

Un Sistema de Gestión de Bases de Datos (DBMS) es un software que permite crear, gestionar y administrar bases de datos. Ejemplos de DBMS populares incluyen MySQL, Oracle, SQL Server y PostgreSQL.

Los Sistemas de Gestión de Bases de Datos (DBMS) son componentes esenciales en el ámbito de la informática, diseñados para facilitar el almacenamiento, gestión y recuperación eficiente de datos. Su propósito principal es actuar como un intermediario entre los usuarios y la base de datos, proporcionando una interfaz que permite a los usuarios definir, consultar y manipular datos sin tener que lidiar directamente con la complejidad del almacenamiento y la recuperación de datos. Los DBMS juegan un

papel crucial en la organización y seguridad de la información, garantizando la integridad y consistencia de los datos almacenados.

Imagen 8: Evolución de los DBMS



Fuente: https://bookdown.org/paranedagarcia/database/normalizacion.html

Selección de un DBMS

Al elegir un DBMS, es importante considerar las necesidades específicas de una aplicación. Cada DBMS tiene sus propias características y ventajas, por lo que la elección debe basarse en los requisitos del proyecto.

Imagen 9: Ejemplos de DBMS



Fuente: https://bookdown.org/paranedagarcia/database/normalizacion.html

Bases de Datos Relacionales

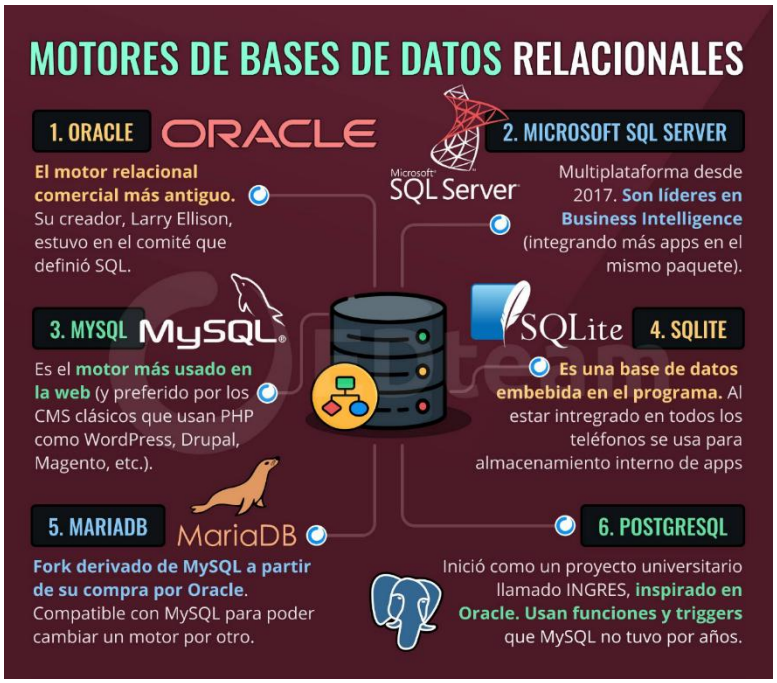
Las bases de datos relacionales son un tipo común de base de datos que utiliza tablas para organizar la información. Cada tabla tiene una clave primaria que identifica de manera única cada registro.

Las Bases de Datos Relacionales (RDBMS) son un tipo fundamental de sistema de gestión de bases de datos que se basa en el modelo relacional, introducido por Edgar F. Codd en la década de 1970. En este modelo, los datos se organizan en tablas con filas y columnas, donde cada fila representa un registro y cada columna representa un atributo o campo específico. Las relaciones entre las tablas se establecen mediante claves primarias y claves foráneas, permitiendo una representación estructurada de datos que es coherente y eficiente.

Principios Fundamentales de las Bases de Datos Relacionales

El modelo relacional se rige por principios fundamentales que lo hacen altamente robusto y versátil. La integridad de los datos se mantiene a través de la normalización, un proceso que reduce la redundancia de datos y evita anomalías en la información. La estructura relacional permite la realización de consultas complejas utilizando lenguajes como SQL (Structured Query Language), lo que facilita la recuperación de datos de manera eficiente y flexible. Además, las bases de datos relacionales son altamente escalables y se utilizan en una amplia gama de aplicaciones, desde sistemas de gestión empresarial hasta aplicaciones web y más. Ejemplos de sistemas de gestión de bases de datos relacionales incluyen MySQL, Oracle, Microsoft SQL Server y PostgreSQL. Estas bases de datos se utilizan en diversos campos, desde el comercio electrónico y la banca hasta la gestión de inventarios y la atención médica. Su importancia radica en su capacidad para proporcionar estructuras de datos coherentes y seguras, lo que garantiza la integridad y la confiabilidad de la información almacenada. Las bases de datos relacionales siguen siendo una pieza fundamental en la gestión de datos y forman la base de sistemas de información críticos en el mundo actual.

Imagen 10: Motores de Base de Datos



Fuente: https://ed.team/blog/6-motores-de-bases-de-datos-sql

SQL (Structured Query Language)

SQL es un lenguaje utilizado para comunicarse con bases de datos relacionales. Permite realizar consultas y manipular datos de manera efectiva.

SQL, que significa Structured Query Language (Lenguaje de Consulta Estructurado), es un lenguaje de programación diseñado para la gestión y manipulación de bases de datos relacionales. Fue desarrollado a fines de la década de 1970 y ha evolucionado hasta convertirse en un estándar en la gestión de datos. SQL proporciona una manera eficaz y uniforme de interactuar con bases de datos, permitiendo a los usuarios crear, consultar, modificar y administrar datos de manera eficiente.

Funciones Clave de SQL

SQL consta de un conjunto de comandos y cláusulas que permiten realizar una amplia gama de operaciones en bases de datos relacionales. Algunas de las funciones clave incluyen la capacidad de recuperar datos utilizando la declaración SELECT, que permite especificar condiciones de búsqueda, orden y agrupación. Además, SQL permite realizar modificaciones en los datos mediante las declaraciones INSERT, UPDATE y DELETE, lo que facilita la actualización de registros. También se utiliza

para definir y modificar la estructura de la base de datos, como crear tablas, definir restricciones y establecer relaciones entre tablas.

Imagen 11: Funciones de SQL



Fuente: https://www.ticportal.es/glosario-tic/base-datos-sql

Importancia y Uso de SQL

SQL desempeña un papel esencial en la gestión de bases de datos relacionales y se utiliza en una variedad de aplicaciones y sistemas, desde aplicaciones empresariales hasta sitios web y sistemas de informes. La importancia de SQL radica en su capacidad para proporcionar una interfaz estandarizada y poderosa que permite a los usuarios y aplicaciones interactuar con bases de datos de manera efectiva. La habilidad para escribir consultas SQL y comprender su funcionamiento es una habilidad fundamental para administradores de bases de datos, desarrolladores de software y profesionales de análisis de datos en el mundo actual. SQL sigue siendo un lenguaje esencial en el ámbito de la informática y la gestión de datos.

Características Principales de las Base de Datos

A continuación, se detallan las características principales de los motores de base de datos:

Seguridad de las Bases de Datos

La seguridad de las bases de datos es fundamental. Los DBMS ofrecen mecanismos de seguridad, incluyendo la autenticación y el control de acceso, para proteger la información.

Integridad de los Datos

Mantener la integridad de los datos es esencial. Las bases de datos utilizan restricciones y reglas para garantizar que los datos sean precisos y coherentes.

Escalabilidad

Un buen diseño de base de datos debe permitir la escalabilidad, es decir, el crecimiento del sistema para adaptarse a las necesidades cambiantes.

Diseño de Bases de Datos

El diseño de bases de datos es un proceso que involucra la creación de una estructura eficiente para almacenar y acceder a datos. Un diseño sólido es esencial para el rendimiento y la fiabilidad de la base de datos.

Rol de las Bases de Datos en la Tecnología de la Información

Las bases de datos son fundamentales en aplicaciones de software, desde sistemas de gestión empresarial hasta aplicaciones web y móviles. Comprender estos conceptos es esencial para cualquier persona que trabaje en el campo de la tecnología de la información.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Base de datos que conozcas, descríbelas**.**

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Tema: Identificar motores de bases de datos y detallar la mayor característica que poseen.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.

2. Observa el video de la clase magistral.

3. Revisa en el calendario el día y hora en que se debe de realizar el Test.

4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

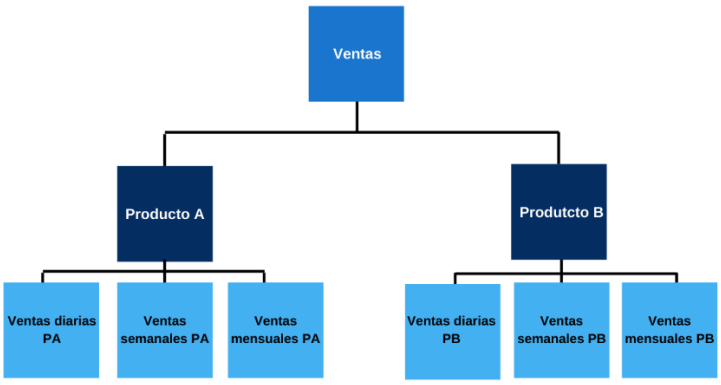
5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

Tema 2: Modelado de Datos Lógicos de Base de datos

Objetivo: Analizar los principios del modelado de datos lógicos en bases de datos, comprendiendo su estructura, componentes y herramientas de diseño, con el propósito de optimizar la organización, integridad y eficiencia en la gestión de la información dentro de los sistemas de bases de datos.

El modelado de datos lógicos de bases de datos es una etapa crucial en el proceso de diseño de sistemas de bases de datos. En esta fase, se representa la estructura y organización de los datos de manera independiente de la implementación física, lo que facilita la comprensión y la gestión de la información. Esta fundamentación teórica explorará en profundidad el modelado de datos lógicos, incluyendo sus conceptos, clasificaciones, ejemplos, historia y componentes clave.

Imagen 1. Ejemplo de Modelado de un escenario



Fuente: https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/

En el emocionante mundo de las bases de datos, el Modelado de Datos Lógicos desempeña un papel crucial en la creación de estructuras que permiten organizar y gestionar la información de manera eficiente. Este proceso se centra en la traducción de los conceptos abstractos del diseño conceptual y el modelo Entidad-Relación a una representación más técnica y específica. Aquí, cada entidad se convierte en una tabla, cada atributo en un campo, y las relaciones se establecen mediante claves foráneas. El objetivo es crear una representación precisa y coherente de la realidad que facilita la manipulación y consulta de datos en el ámbito de los sistemas de gestión de bases de datos.

En este proceso, nos valdremos de diversas herramientas y lenguajes especializados para plasmar el diseño conceptual en una estructura lógica. SQL (Structured Query Language) desempeña un papel crucial, permitiéndonos definir tablas, establecer relaciones y garantizar la integridad de los datos. Herramientas de modelado como Oracle SQL Developer Data Modeler, Erwin Data Modeler o MySQL Workbench nos

ofrecen interfaces visuales para diseñar y representar gráficamente la estructura de la base de datos, simplificando la creación y comprensión de modelos lógicos complejos.

Imagen 2. Herramientas para el modelado de una base de datos



Fuente: https://www.proavan.com/es/herramientas-libres-para-modelado-sobre-mysql/

Importancia del Modelado de Datos Lógicos en el Desarrollo de Software

El Modelado de Datos Lógicos no solo es un paso intermedio en el proceso de diseño de bases de datos, sino que también desempeña un papel crucial en el desarrollo de software. Proporciona la base sobre la cual los desarrolladores construyen aplicaciones que interactúan con la base de datos, asegurando la coherencia y la integridad en el manejo de datos. Además, facilita la identificación temprana de posibles problemas en el diseño, permitiendo ajustes antes de la implementación final. En resumen, el Modelado de Datos Lógicos es una herramienta esencial que contribuye significativamente al desarrollo de sistemas robustos y eficientes en el vasto mundo de la gestión de datos.

Conceptualizaciones:

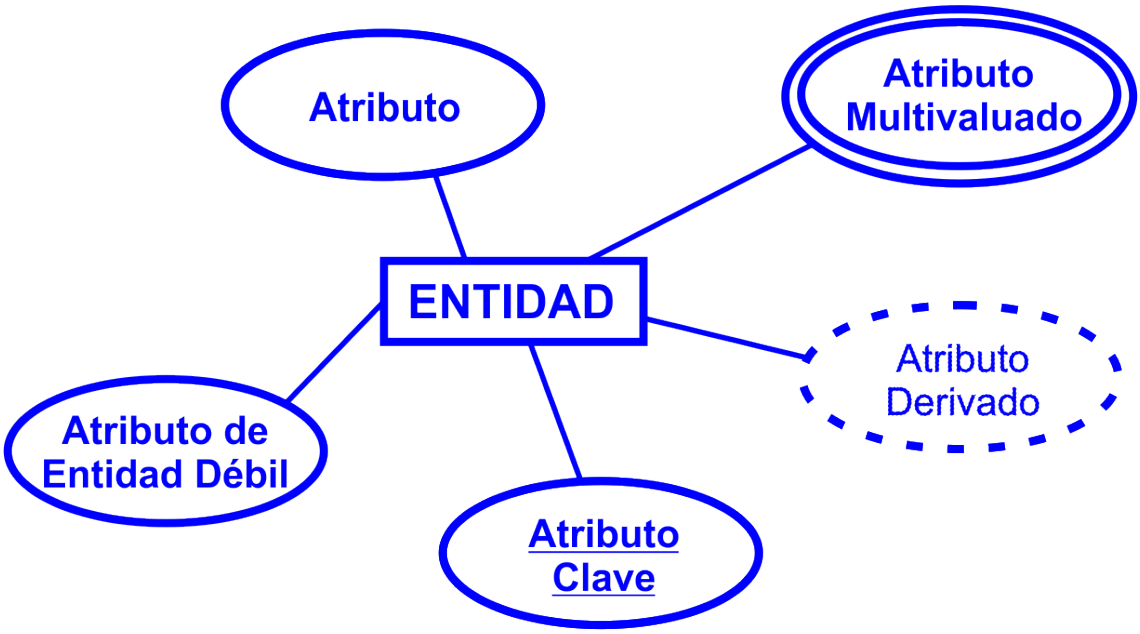
El modelado de datos lógicos es un enfoque que utiliza modelos conceptuales para representar la estructura de la información en una base de datos. Estos modelos describen entidades, atributos y relaciones entre los datos. Los conceptos fundamentales incluyen:

Entidades: Son objetos o conceptos del mundo real que se representan en la base de datos, como "clientes" o "productos".

Atributos: Son características o propiedades de las entidades que describen los datos, como el nombre del cliente o el precio del producto.

Relaciones: Definen cómo las entidades se relacionan entre sí, como la relación entre "cliente" y "pedido".

Imagen 3. Entidades y Atributos en el modelado de una base de datos



Fuente: https://www.abcsoftperu.com/descargas/curso02.html

Clasificaciones:

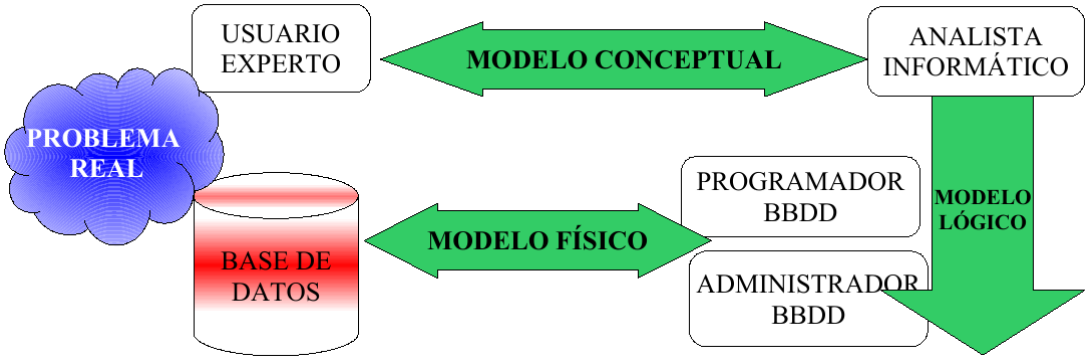
El modelado de datos lógicos se divide en varias clasificaciones, siendo las más destacadas:

Modelo de Entidad-Relación (ER): Este modelo se centra en la representación de entidades y sus relaciones. Utiliza diagramas de entidad-relación para visualizar la estructura de datos.

Modelo Relacional: Representa los datos en tablas, con relaciones establecidas a través de claves primarias y foráneas. SQL es un lenguaje común para trabajar con bases de datos relacionales.

Modelo de Objetos: Enfocado en la representación de objetos y sus relaciones, este modelo se utiliza en bases de datos orientadas a objetos.

Imagen 4. Clasificación de modelos



Fuente: https://gestionbasesdatos.readthedocs.io/es/latest/Tema2/Teoria.html

Ejemplos:

Un ejemplo sencillo de modelado de datos lógicos sería una base de datos para una tienda en línea. Las entidades podrían incluir "producto" y "cliente", con atributos como "nombre" y "precio". La relación "realiza" podría conectar un cliente con un pedido. Los diagramas de entidad-relación y las tablas relacionales se utilizarían para representar esta estructura.

Componentes:

Los componentes clave del modelado de datos lógicos incluyen:

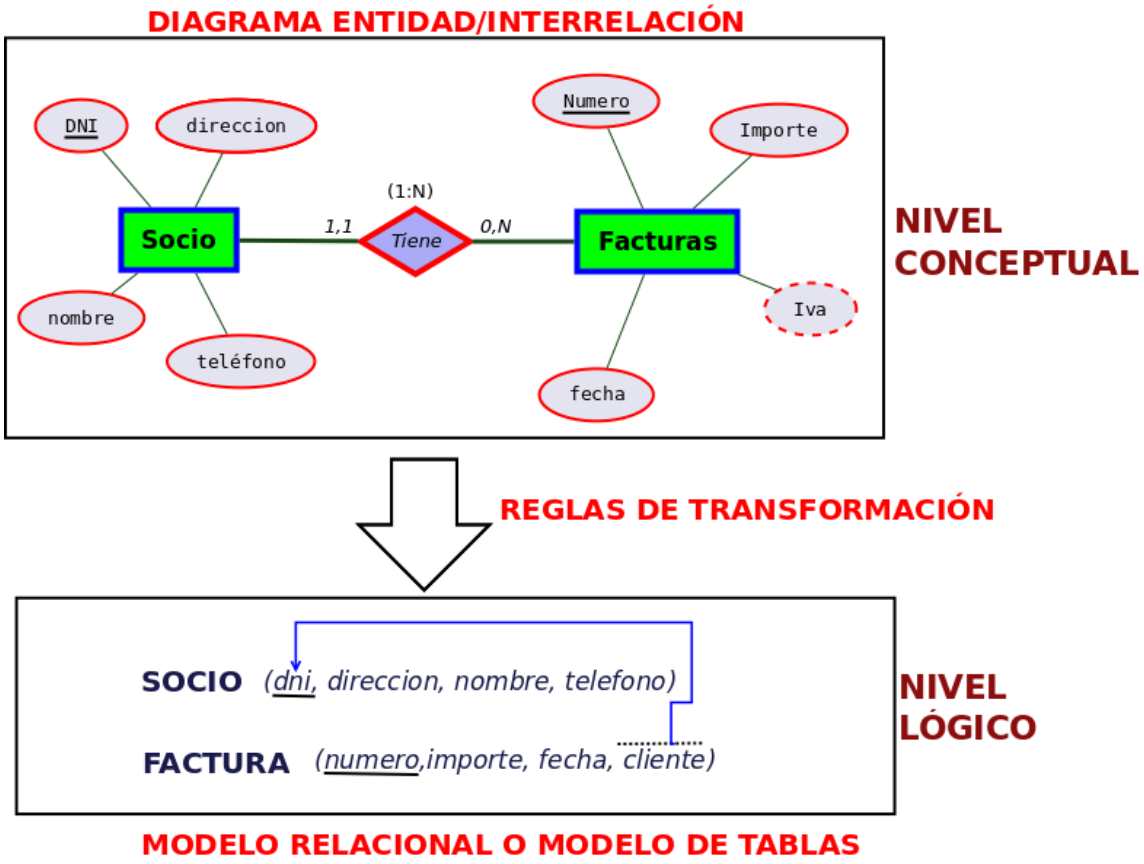
Diagramas de entidad-relación: Estos diagramas visualizan entidades, atributos y relaciones en un formato gráfico que facilita la comprensión.

Tablas y campos: En el modelo relacional, las tablas y los campos representan la estructura de datos de manera física y lógica.

Claves primarias y foráneas: Establecen relaciones entre las tablas y garantizan la integridad de los datos.

Normalización: Es el proceso de organizar los datos para eliminar la redundancia y mejorar la integridad.

Imagen 5. Componentes de modelos



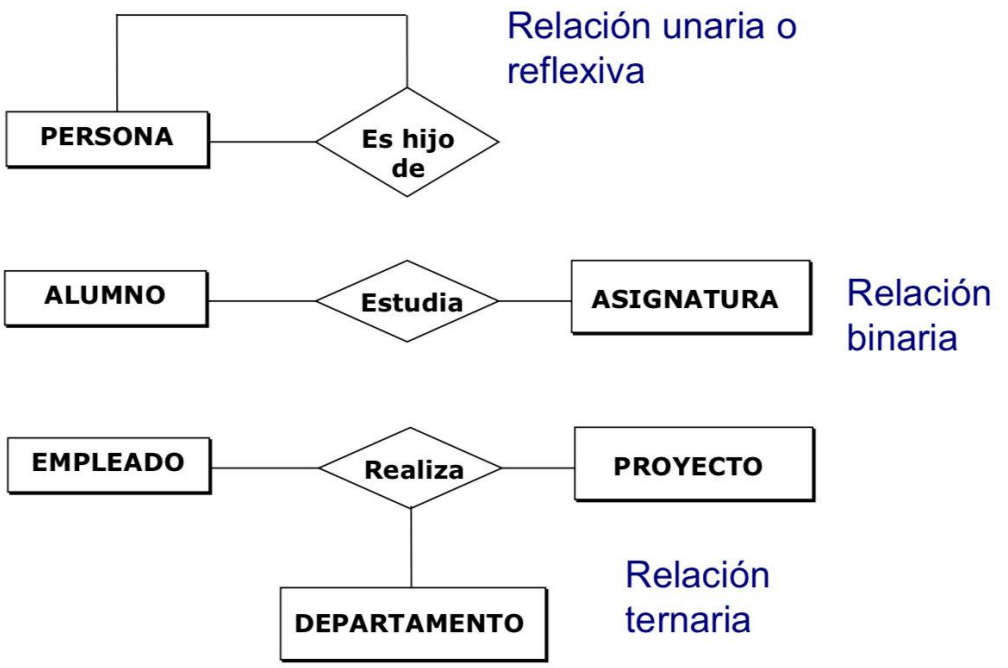
Fuente: https://es.wikieducator.org/Plantilla:Bases_Datos/Modelo_Logico

El modelado de datos lógicos es una parte esencial en el diseño de bases de datos eficientes y coherentes. Permite a los diseñadores definir la estructura de datos de manera independiente de la implementación física, lo que facilita el mantenimiento y la comprensión de la información almacenada. Además, su historia rica y su evolución a lo largo de las décadas reflejan su importancia continua en el campo de la gestión de datos.

Modelo Conceptual de Base de Datos.

En el ámbito de las bases de datos, el Modelo Conceptual se presenta como la primera etapa del proceso de diseño, desempeñando un papel fundamental en la creación de un plan general para organizar la información. Este modelo se enfoca en la representación abstracta de los elementos clave, como entidades, atributos y las relaciones entre ellas. Es un punto de partida esencial para comprender la naturaleza de los datos que se manejarán y establecer la base sobre la cual se construirá la estructura lógica y física de la base de datos. Al capturar la esencia de la realidad que se desea modelar, el Modelo Conceptual brinda una visión clara y comprensible de la información que será gestionada.

Imagen 6. Modelo Conceptual



Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html

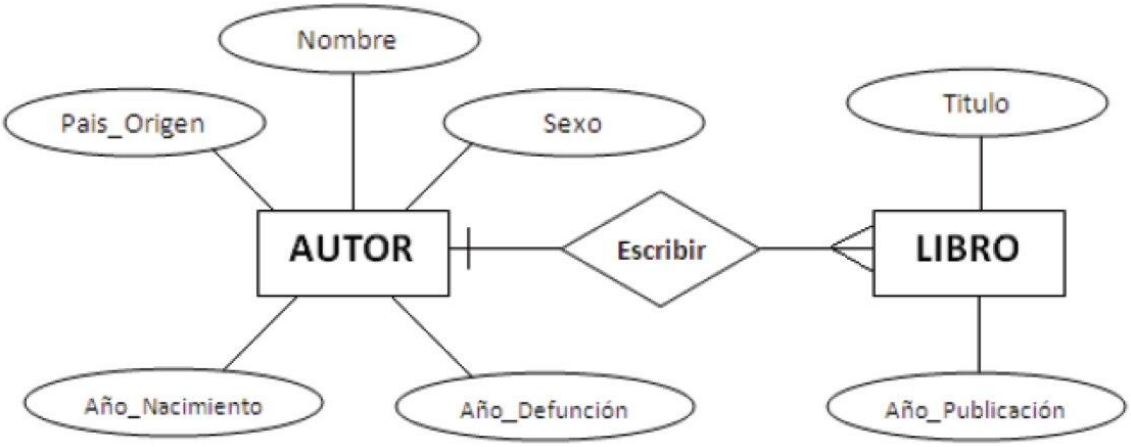
Elementos Clave del Modelo Conceptual

Dentro del Modelo Conceptual, se destacan tres elementos principales: entidades, atributos y relaciones. Las entidades representan objetos o conceptos distintos del mundo real, como clientes, productos o empleados. Los atributos son las características específicas de cada entidad, describiendo aspectos como nombre, dirección o fecha de nacimiento. Las relaciones definen las conexiones y dependencias entre entidades, permitiendo reflejar cómo interactúan y se asocian entre sí. Estos elementos se representan gráficamente utilizando herramientas como el diagrama de entidad-relación (ER), que proporciona una visualización clara y comprensible de la estructura conceptual.

Diseño de Entidades y Atributos

Una parte esencial del Modelo Conceptual implica el diseño detallado de entidades y atributos. Aquí, se definen las características únicas de cada entidad y se especifican los atributos necesarios para describirlas de manera completa. Por ejemplo, en un contexto de gestión de empleados, una entidad "Empleado" podría tener atributos como "Nombre", "Número de Empleado" y "Fecha de Contratación". El proceso de diseño de entidades y atributos es crucial para capturar la información relevante y garantizar que el modelo conceptual refleje con precisión la realidad que se desea representar.

Imagen 7. Modelo Conceptual – Atributos – Entidades - Relaciones



Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html

Las relaciones entre entidades son uno de los aspectos más críticos del Modelo Conceptual. Estas conexiones determinan cómo la información se entrelaza y cómo se pueden recuperar datos de manera eficiente. Por ejemplo, en un modelo conceptual para una tienda en línea, la entidad "Cliente" podría tener una relación con la entidad "Pedido", estableciendo cómo se relacionan las compras de los clientes con sus perfiles. El diseño preciso de estas relaciones garantiza una representación fiel de la dinámica del mundo real que estamos modelando.

Transición a los Modelos Lógico y Físico

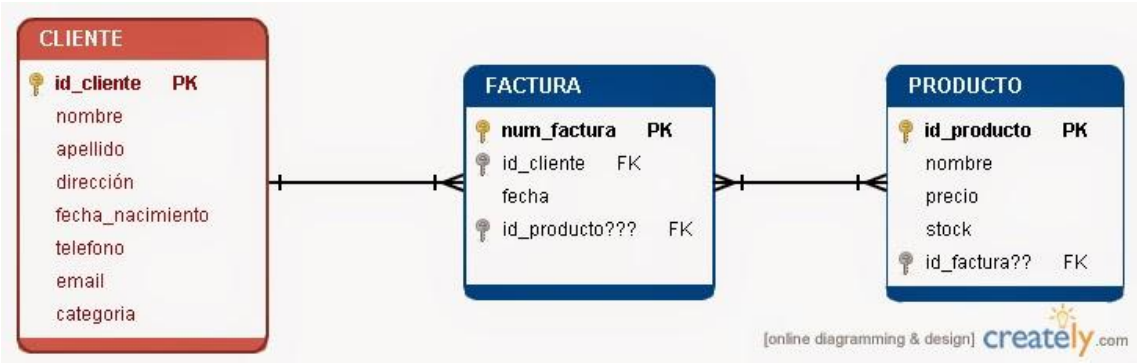
El Modelo Conceptual sirve como el punto de partida para el desarrollo posterior del diseño de la base de datos. Después de su creación, se pasa a los modelos lógico y físico, donde se transforma la representación abstracta en una estructura técnica y específica. Durante esta transición, se utilizan herramientas como SQL para definir las tablas y relaciones concretas. En resumen, el Modelo Conceptual es el cimiento esencial en el diseño de bases de datos, proporcionando una visión clara y comprensible que guía el desarrollo subsiguiente de estructuras más detalladas.

Modelo Entidad-Relación (ER)

El Modelo Entidad-Relación (ER) es una herramienta gráfica que desempeña un papel esencial en el diseño de bases de datos al proporcionar una representación visual y clara de la estructura conceptual. Este modelo se centra en tres elementos clave: entidades, atributos y relaciones. Las entidades representan objetos o conceptos del mundo real, como "Cliente" o "Producto". Los atributos describen las características específicas de cada entidad, como "Nombre" o "Precio". Las relaciones definen las

asociaciones y conexiones entre entidades, permitiendo capturar la dinámica de cómo interactúan los diferentes elementos.

Imagen 8. Modelo Entidad Relación



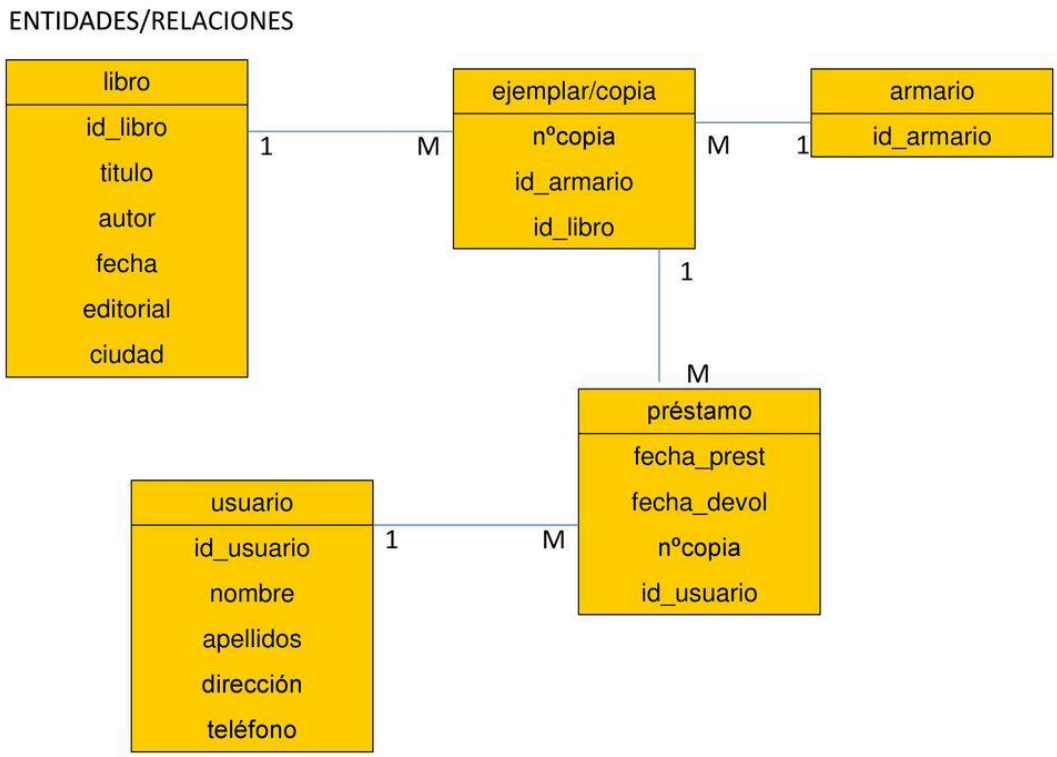
Fuente: https://www.develou.com/sistema-facturacion-base-datos/

Elementos Claves del Modelo ER con Ejemplos

En el Modelo ER, las entidades se representan mediante rectángulos, los atributos con elipses y las relaciones con líneas conectivas. Por ejemplo, en el contexto de una base de datos para una biblioteca, tendríamos una entidad "Libro" con atributos como "Título" y "Autor". La relación entre "Libro" y "Autor" se establecería para indicar que un autor puede escribir varios libros. Este enfoque visual simplifica la comprensión y comunicación del diseño conceptual. El Modelo ER se convierte así en una herramienta poderosa para traducir las ideas abstractas del diseño conceptual a una representación gráfica clara y comprensible.

En el diseño de entidades y atributos dentro del Modelo ER, se busca identificar y definir las características únicas de cada entidad. Siguiendo con el ejemplo de una base de datos para una tienda en línea, la entidad "Producto" podría tener atributos como "ID de Producto", "Nombre" y "Precio". La representación gráfica permite una visualización rápida y precisa de la información que estamos modelando, facilitando la toma de decisiones sobre la estructura final de la base de datos.

Imagen 9. Modelo Entidad Relación - Librería



Fuente: https://slideplayer.es/slide/13182238/

Las relaciones son un componente crítico en el Modelo ER, ya que capturan la dinámica de cómo interactúan las diferentes entidades. Por ejemplo, en un sistema de gestión de empleados, la entidad "Departamento" podría tener una relación con la entidad "Empleado" para indicar que un departamento puede tener varios empleados. La representación visual de estas relaciones aporta claridad y comprensión al diseño, permitiendo a los diseñadores y usuarios entender fácilmente la conexión entre los distintos elementos de la base de datos.

Transición del Modelo ER al Diseño Lógico y Físico

Después de haber desarrollado un Modelo ER sólido, la transición hacia el diseño lógico y físico implica convertir esta representación conceptual en una estructura técnica y específica. Herramientas como SQL se utilizan para traducir las entidades, atributos y relaciones en tablas y claves en el mundo de los sistemas de gestión de bases de datos. En resumen, el Modelo ER es una fase esencial en el diseño de bases de datos, proporcionando una representación gráfica que sirve como base para el desarrollo subsiguiente de la estructura detallada de la base de datos.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Modelo de Base de datos de un hospital.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Crear un modelo de bases de Datos MER.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.

2. Observa el video de la clase magistral.

3. Revisa en el calendario el día y hora en que se debe de realizar el Test.

4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

II. DATOS GENERALES DE LA UNIDAD 2

Información General

1.1.	Nivel	Segundo					
1.2.	Nombre Unidad	Aplicación de Modelos de Base de Datos.					
1.3.	Tema 1 – Unidad	Modelo Entidad Relación.					
1.4.	Tema 2 – Unidad	Sistemas de Gestión de Base de Datos.					
1.5.	Unidad organizacional	Unidad Profesional					
1.6.	Unidad	2					
1.7.	Total, Horas Unidad	15					
1.8.	Detalle de horas Unidad	ACD	6	AA	4	APE	5

Resultado de Aprendizaje

Al finalizar la unidad, el estudiante comprenderá los fundamentos del Modelo Entidad-Relación (MER), identificando sus componentes y aplicándolo en el diseño lógico de bases de datos. Además, adquirirá conocimientos sobre los Sistemas de Gestión de Bases de Datos (SGBD), explorando sus características, funciones y aplicaciones en distintos entornos. Desarrollará habilidades para diseñar, implementar y administrar bases de datos eficientes, asegurando la integridad y optimización en la gestión de la información.

Metodología

El aprendizaje se desarrollará mediante el análisis de lecturas interactivas, videos explicativos y estudios de caso sobre el diseño y administración de bases de datos. Se fomentará la aplicación práctica a través de la elaboración de diagramas Entidad-Relación utilizando herramientas como MySQL Workbench, Lucidchart o Draw.io. Para fortalecer el manejo de SGBD, se realizarán ejercicios prácticos en plataformas como MySQL, PostgreSQL o SQL Server, donde los estudiantes ejecutarán consultas, creaciones de tablas y manipulaciones de datos. La retroalimentación docente y el trabajo colaborativo en foros facilitarán la resolución de dudas y la optimización de los modelos de bases de datos diseñados.

DESARROLLO

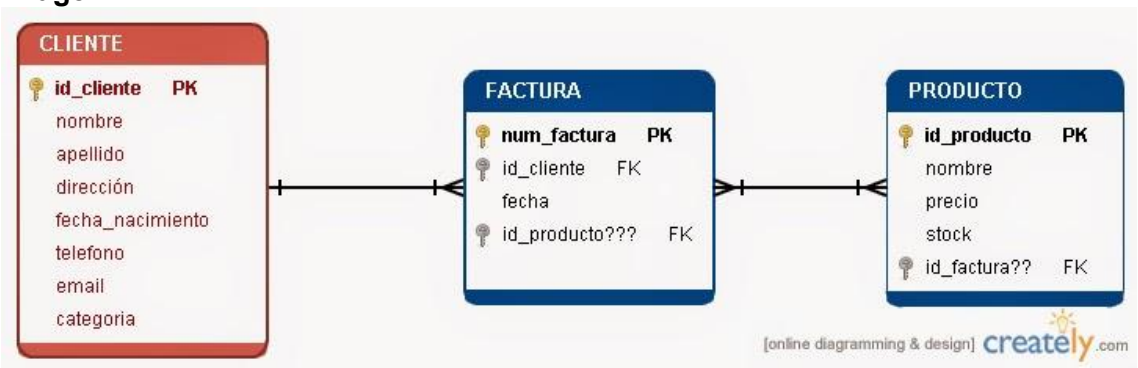
Unidad 2: Aplicación de Modelos de Base de Datos

Tema 1: Modelo Entidad Relación

Objetivo: Comprender y aplicar el Modelo Entidad-Relación (MER) en el diseño de bases de datos, analizando sus elementos, reglas y notaciones, con el propósito de estructurar sistemas de información eficientes y garantizar la integridad de los datos.

El Modelo Entidad-Relación (ER) es una herramienta gráfica que desempeña un papel esencial en el diseño de bases de datos al proporcionar una representación visual y clara de la estructura conceptual. Este modelo se centra en tres elementos clave: entidades, atributos y relaciones. Las entidades representan objetos o conceptos del mundo real, como "Cliente" o "Producto". Los atributos describen las características específicas de cada entidad, como "Nombre" o "Precio". Las relaciones definen las asociaciones y conexiones entre entidades, permitiendo capturar la dinámica de cómo interactúan los diferentes elementos.

Imagen 1. Modelo Entidad Relación



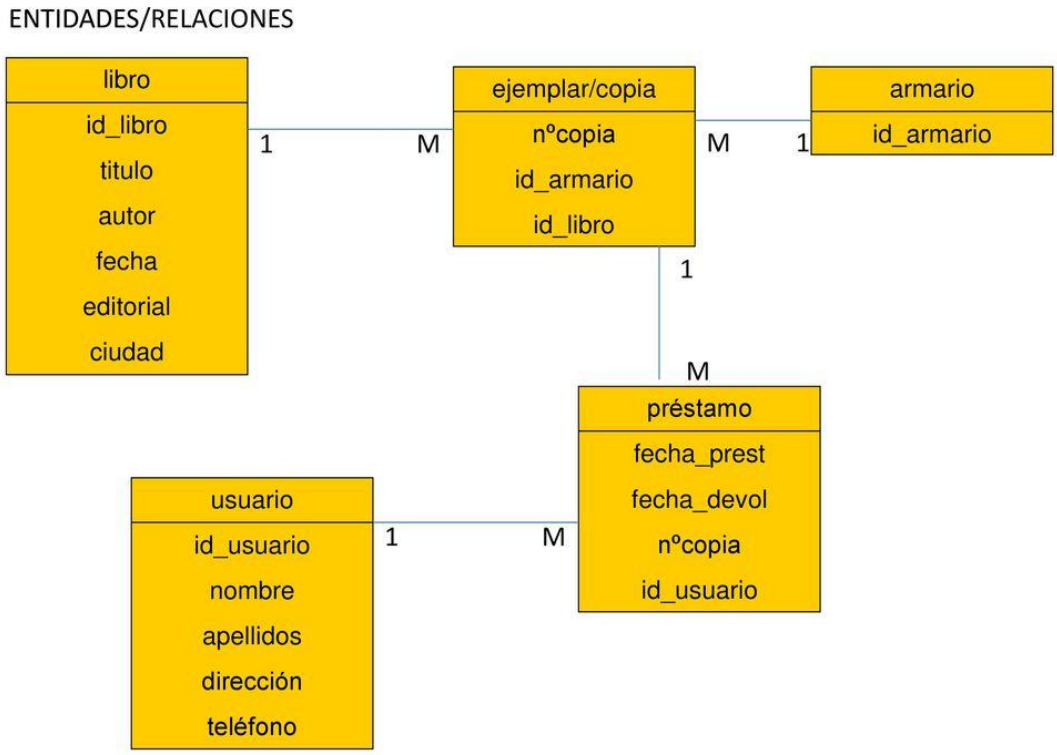
Fuente: https://www.develou.com/sistema-facturacion-base-datos/

Elementos Claves del Modelo ER con Ejemplos

En el Modelo ER, las entidades se representan mediante rectángulos, los atributos con elipses y las relaciones con líneas conectivas. Por ejemplo, en el contexto de una base de datos para una biblioteca, tendríamos una entidad "Libro" con atributos como "Título" y "Autor". La relación entre "Libro" y "Autor" se establecería para indicar que un autor puede escribir varios libros. Este enfoque visual simplifica la comprensión y comunicación del diseño conceptual. El Modelo ER se convierte así en una herramienta poderosa para traducir las ideas abstractas del diseño conceptual a una representación gráfica clara y comprensible.

En el diseño de entidades y atributos dentro del Modelo ER, se busca identificar y definir las características únicas de cada entidad. Siguiendo con el ejemplo de una base de datos para una tienda en línea, la entidad "Producto" podría tener atributos como "ID de Producto", "Nombre" y "Precio". La representación gráfica permite una visualización rápida y precisa de la información que estamos modelando, facilitando la toma de decisiones sobre la estructura final de la base de datos.

Imagen 2. Modelo Entidad Relación - Librería



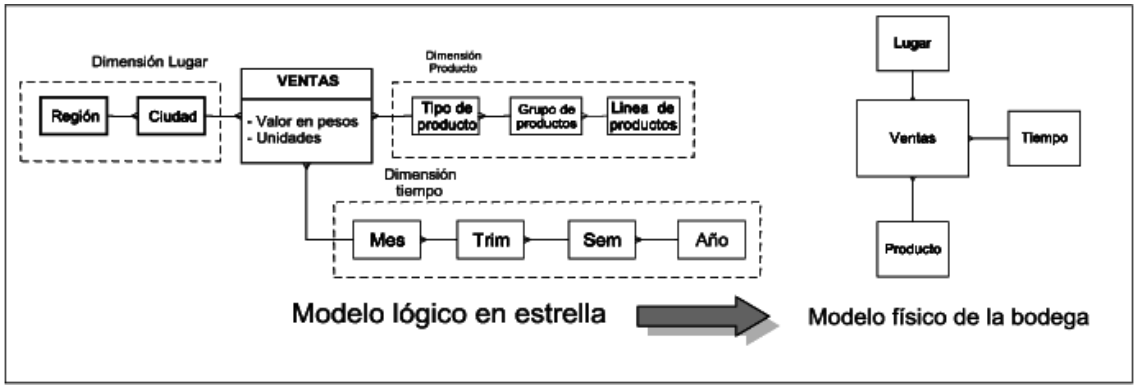
Fuente: https://slideplayer.es/slide/13182238/

Las relaciones son un componente crítico en el Modelo ER, ya que capturan la dinámica de cómo interactúan las diferentes entidades. Por ejemplo, en un sistema de gestión de empleados, la entidad "Departamento" podría tener una relación con la entidad "Empleado" para indicar que un departamento puede tener varios empleados. La representación visual de estas relaciones aporta claridad y comprensión al diseño, permitiendo a los diseñadores y usuarios entender fácilmente la conexión entre los distintos elementos de la base de datos.

Transición del Modelo ER al Diseño Lógico y Físico

Después de haber desarrollado un Modelo ER sólido, la transición hacia el diseño lógico y físico implica convertir esta representación conceptual en una estructura técnica y específica. Herramientas como SQL se utilizan para traducir las entidades, atributos y relaciones en tablas y claves en el mundo de los sistemas de gestión de bases de datos. En resumen, el Modelo ER es una fase esencial en el diseño de bases de datos, proporcionando una representación gráfica que sirve como base para el desarrollo subsiguiente de la estructura detallada de la base de datos.

Imagen 3. Modelo Lógico a Físico



Fuente: https://www.researchgate.net/figure/Figura-11-Modelo-logico-en-estrella-y-modelo-fisico-de-la-bodega_fig9_38319764/

Historia

La historia de los modelos Entidad-Relación (ER) se remonta a la década de 1970, cuando el Dr. Peter Chen, un científico de la computación, introdujo por primera vez este enfoque conceptual para el diseño de bases de datos. Chen presentó sus ideas en un artículo seminal titulado "The Entity-Relationship Model - Toward a Unified View of Data" en 1976, que fue publicado en la revista "ACM Transactions on Database Systems".

En este artículo, Chen propuso el Modelo Entidad-Relación como una herramienta gráfica para representar las relaciones entre entidades en un sistema de información. Su objetivo era proporcionar un enfoque más intuitivo y comprensible para el diseño de bases de datos al utilizar diagramas que reflejaran la estructura lógica de la información en lugar de depender únicamente de descripciones textuales.

La propuesta de Chen introdujo conceptos clave como entidades, atributos y relaciones. Las entidades representan objetos o conceptos en el mundo real (por ejemplo, un estudiante o un producto), los atributos describen las características de esas entidades, y las relaciones establecen vínculos significativos entre entidades.

A medida que las bases de datos y los sistemas de información crecieron en complejidad, el Modelo ER se consolidó como una herramienta esencial para los diseñadores y desarrolladores de bases de datos. La notación gráfica del Modelo ER proporciona una forma efectiva de visualizar y comunicar la estructura conceptual de la información, facilitando la colaboración entre diseñadores, usuarios y desarrolladores.

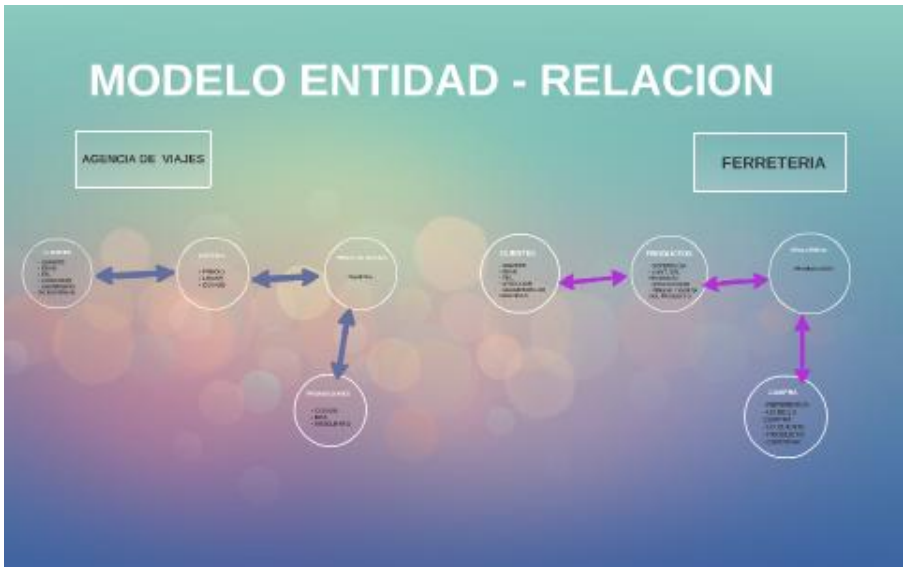
A lo largo de los años, el Modelo ER ha evolucionado y se ha adaptado para abordar desafíos específicos en el diseño de bases de datos. Variantes del modelo, como el Modelo de Datos de Entidad-Relación Extendido (EER), han sido desarrolladas para proporcionar una mayor expresividad y capacidad de representación de situaciones más complejas.

En resumen, la historia de los modelos Entidad-Relación se origina en la visión de Peter Chen de ofrecer un enfoque más visual y comprensible para el diseño de bases de datos, y su contribución ha tenido un impacto significativo en la forma en que conceptualizamos y diseñamos sistemas de información.

Ventajas:

- Claridad Visual: Proporciona una representación visual clara de la estructura conceptual de la base de datos.
- Facilita la Comunicación: Permite a diseñadores y usuarios comunicarse eficientemente sobre la estructura de la base de datos.
- Diseño Intuitivo: Facilita la comprensión y diseño intuitivo de la base de datos, simplificando la fase de conceptualización.

Imagen 5. MER



Fuente: https://abzlocal.mx/introducir-85-imagen-modelo-de-entidad-relacion-de-una-ferreteria/

Desventajas:

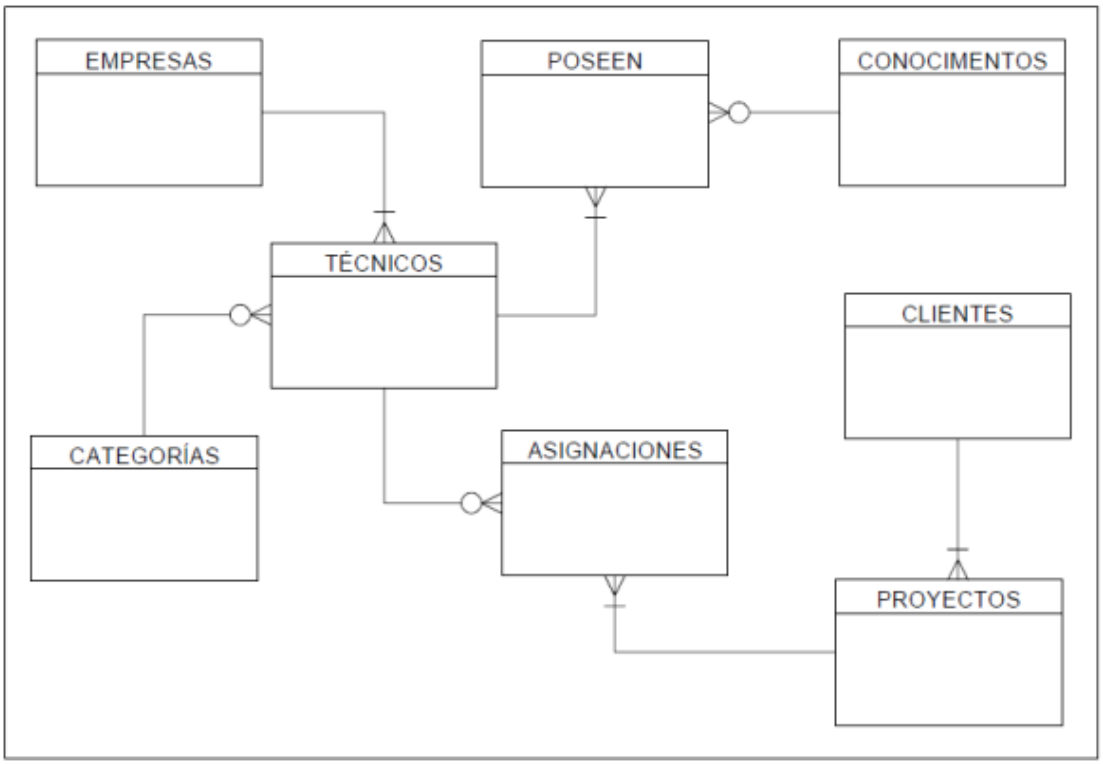
- Simplificación Excesiva: Puede simplificar demasiado la realidad, ya que algunas complejidades del mundo real pueden ser difíciles de representar de manera efectiva.
- Limitaciones en la Expresividad: A veces, puede no ser lo suficientemente expresivo para modelar situaciones complejas de manera precisa.

Importancia del Modelado Entidad Relación

La importancia de los modelos de entidad-relación (ER) en el diseño de bases de datos es fundamental y se deriva de varias razones clave:

Visualización y Claridad Conceptual: Los modelos de entidad-relación ofrecen una representación visual y clara de la estructura conceptual de una base de datos. Utilizando notaciones gráficas, como el diagrama ER, los diseñadores y usuarios pueden comprender de manera intuitiva cómo se relacionan las entidades y cuál es la naturaleza de esas relaciones.

Imagen 6. Visualización y Claridad Conceptual en MER



Fuente: https://gsitic.wordpress.com/2018/04/17/biii6-modelizacion-conceptual-el-modelo-entidad-relacion-extendido-e-r-elementos-reglas-de-modelizacion-validacion-y-construccion-de-modelos-de-datos/

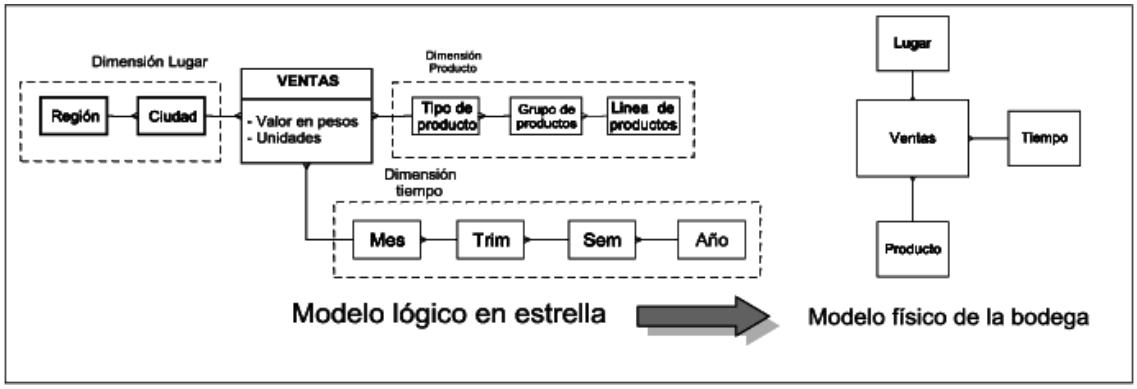
Comunicación Efectiva: Los modelos de entidad-relación facilitan la comunicación entre los diversos interesados en el desarrollo de la base de datos, como diseñadores, desarrolladores y usuarios finales. La notación gráfica simplifica la transmisión de ideas complejas, permitiendo que todos los involucrados tengan una visión común y compartida de la estructura de la base de datos.

Diseño y Planificación Eficientes: Al proporcionar una representación clara de las entidades, atributos y relaciones, los modelos de entidad-relación ayudan en la fase de diseño inicial de una base de datos. Los diseñadores pueden planificar de manera más eficiente cómo organizar y estructurar la información antes de pasar a fases más detalladas del desarrollo.

Identificación de Requisitos y Ambigüedades: Durante el proceso de modelado de entidad-relación, es posible identificar y aclarar los requisitos del sistema de información. La visualización de las relaciones entre entidades puede revelar ambigüedades en los requisitos y ayudar a refinar la comprensión del sistema.

Facilita la Transición a Modelos Lógicos y Físicos: Los modelos de entidad-relación sirven como punto de partida para la transición a modelos lógicos y físicos de bases de datos. Proporcionan una representación conceptual sólida que se traduce de manera efectiva en tablas, claves y relaciones en un sistema de gestión de bases de datos.

Imagen 7. Modelo Lógico a Físico



Fuente: https://www.researchgate.net/figure/Figura-11-Modelo-logico-en-estrella-y-modelo-fisico-de-la-bodega_fig9_38319764/

Adaptabilidad a Cambios: Cuando se realizan cambios en los requisitos del sistema, los modelos de entidad-relación permiten una adaptabilidad más eficiente. Los diseñadores pueden ajustar las entidades y relaciones en el modelo antes de implementar cambios a nivel lógico o físico.

Soporte para Diseño de Bases de Datos Normalizadas: Los modelos de entidad-relación facilitan la aplicación de principios de normalización en el diseño de bases de datos, ayudando a eliminar redundancias y garantizar la consistencia de los datos.

En resumen, los modelos de entidad-relación desempeñan un papel crucial en el ciclo de vida del desarrollo de bases de datos, desde la conceptualización hasta la implementación. Proporcionan una herramienta invaluable para la planificación, diseño y comunicación efectiva en el desarrollo de sistemas de información basados en bases de datos.

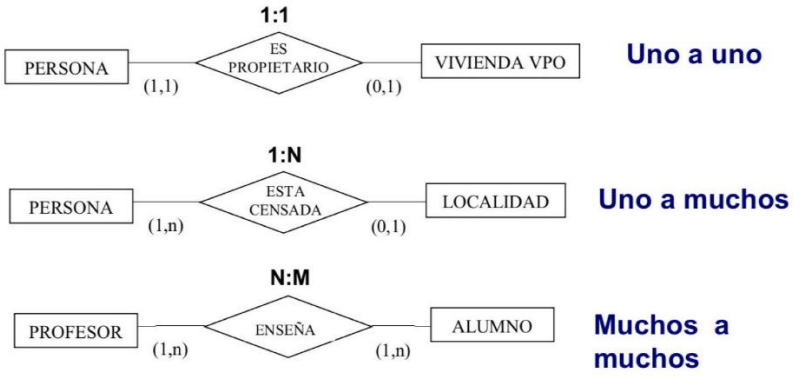
Cardinalidades en los Modelos Entidad Relación

En el Modelo Entidad-Relación (ER), las cardinalidades describen la naturaleza y la cantidad de las relaciones entre las entidades. Hay tres tipos principales de cardinalidades: uno a uno (1:1), uno a muchos (1:N), y muchos a muchos (N:N). Veamos ejemplos de cada uno:

Uno a Uno (1:1):

En una relación uno a uno, cada instancia de una entidad está relacionada con exactamente una instancia de otra entidad y viceversa. Un ejemplo común es la relación entre "Persona" y "Pasaporte". Cada persona tiene un único pasaporte y cada pasaporte está asociado a una única persona.

Imagen 8. Cardinalidades Modelo ER



Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos-1.html

Uno a Muchos (1:N):

En una relación uno a muchos, una instancia de una entidad está relacionada con muchas instancias de otra entidad, pero una instancia de la segunda entidad está relacionada con solo una instancia de la primera entidad. Por ejemplo, la relación entre "Departamento" y "Empleado". Un departamento puede tener muchos empleados, pero cada empleado trabaja en un solo departamento.

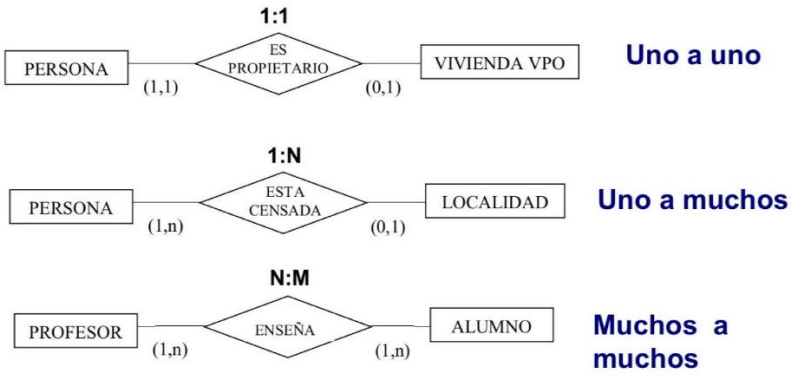


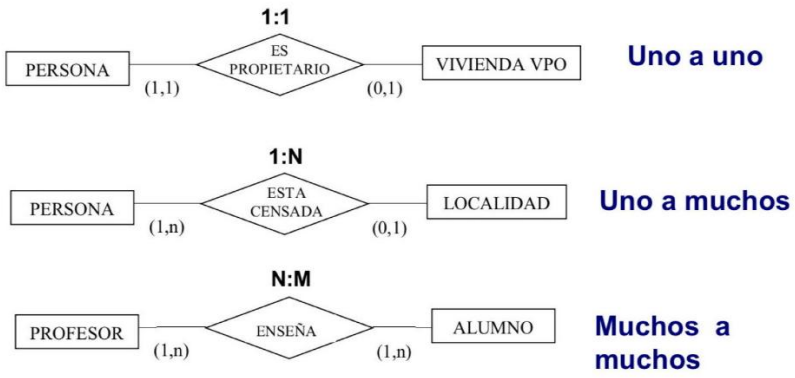
Imagen 9. Cardinalidades Modelo ER

Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos-1.html

Muchos a Muchos (N:N):

En una relación muchos a muchos, muchas instancias de una entidad están relacionadas con muchas instancias de otra entidad. Para representar esto, se utiliza una entidad intermedia o tabla de asociación. Por ejemplo, la relación entre "Estudiante" y "Curso". Un estudiante puede estar inscrito en muchos cursos, y un curso puede tener muchos estudiantes.

Imagen 10. Cardinalidades Modelo ER



Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos-1.html

Modelado de datos físicos de base de datos

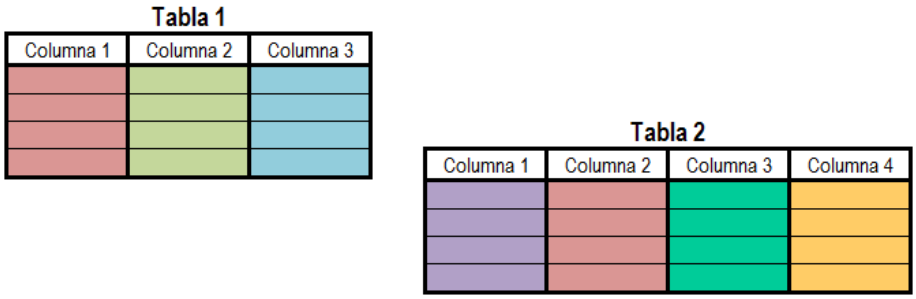
El modelado de datos físicos de una base de datos se refiere a la representación técnica y detallada de la estructura de la base de datos en un nivel que puede ser implementado directamente en un sistema de gestión de bases de datos (DBMS). A diferencia del modelado conceptual o lógico, que se enfoca en la representación de la información de manera abstracta, el modelado físico considera aspectos más técnicos y específicos relacionados con la implementación concreta en el entorno de un sistema de bases de datos.

Aquí hay algunos aspectos clave del modelado de datos físicos:

Definición de Tablas y Columnas:

- Identificación de las tablas necesarias para almacenar la información.
- Definición de columnas con tipos de datos específicos para cada atributo.

Imagen 11. Tablas y Columnas

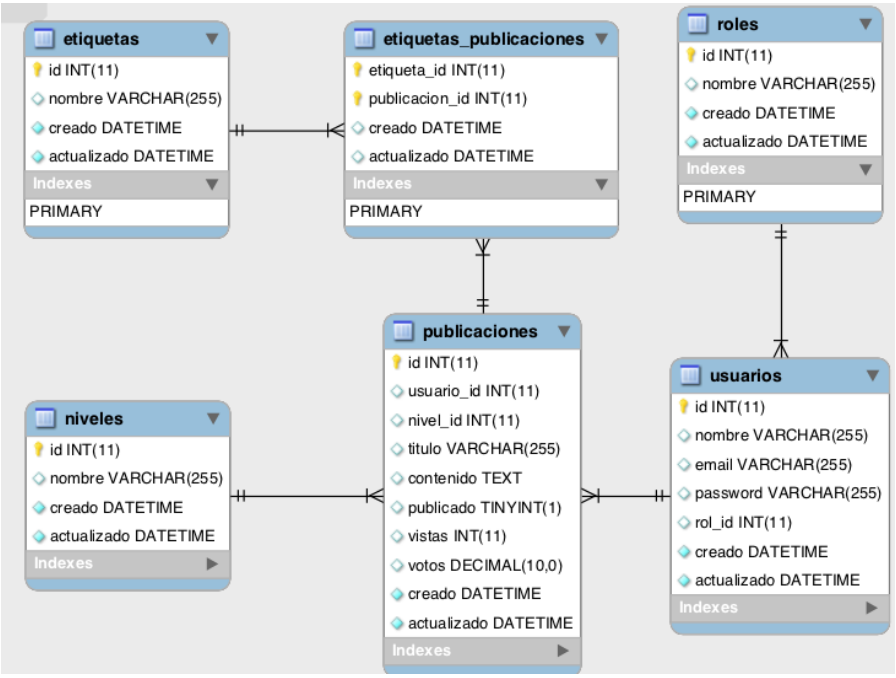


Fuente: https://bookdown.org/paranedagarcia/database/modelamiento-de-datos-1.html

Claves y Restricciones:

- Especificación de claves primarias y foráneas para garantizar la integridad referencial.
- Aplicación de restricciones, como restricciones de unicidad o de verificación.

Imagen 12. Claves y restricciones

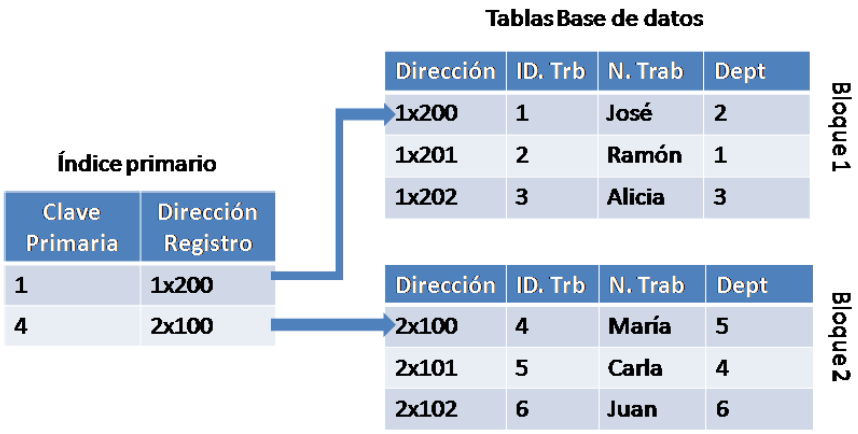


Fuente: https://blog.eamexicano.com/mysql/integridad-referencial/

Índices:

- Creación de índices para mejorar el rendimiento de las consultas.
- Selección de las columnas más relevantes para la creación de índices basados en los patrones de acceso a los datos.

Imagen 13. Indices



Fuente:
https://es.m.wikibooks.org/wiki/Archivo:Ejemplo_Indice_primario_%28Bases_de_datos_relacionales%29.png

Particionamiento:

- División de grandes conjuntos de datos en particiones más pequeñas para mejorar la administración y el rendimiento.

Espacio de Almacenamiento:

- Especificación del espacio de almacenamiento para cada tabla y sus objetos asociados (índices, vistas, etc.).
- Gestión de tablespaces y asignación de tamaños de archivo.

Optimización de Consultas:

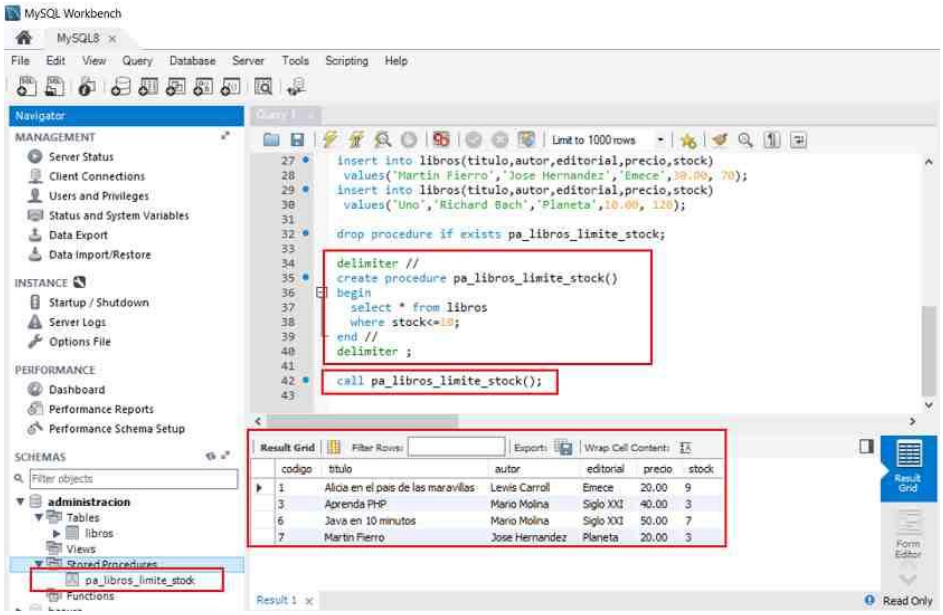
- Consideración de estrategias de optimización, como el uso de vistas materializadas o funciones analíticas.
- Evaluación de planes de ejecución de consultas para mejorar la eficiencia.

Procedimientos Almacenados y Triggers:

- Creación de procedimientos almacenados para ejecutar lógica de negocio en el lado del servidor.

- Definición de triggers para responder a eventos específicos, como inserciones, actualizaciones o eliminaciones.

Imagen 13. Procedimientos almacenados



Fuente:
https://www.tutorialesprogramacionya.com/mysqlya/temarios/descripcion.php?inicio=84&cod=102&punto=99

Seguridad y Privilegios:

- Asignación de privilegios y permisos a usuarios y roles.
- Implementación de políticas de seguridad para restringir el acceso a los datos sensibles.

El modelado de datos físicos transforma el diseño conceptual y lógico en una estructura técnica que puede ser implementada en un DBMS específico. Es crucial para garantizar un rendimiento óptimo, una administración eficiente y la integridad de los datos en el entorno de producción de la base de datos.

Introducción Instalación del sistema de base de datos

La instalación de MySQL y MySQL Workbench es el primer paso esencial para aquellos que desean gestionar bases de datos utilizando el sistema de gestión de bases de datos MySQL. A continuación, se detallan los pasos para llevar a cabo la instalación, desde la descarga hasta la configuración inicial.

Pasos para la Instalación:

Descarga de MySQL Server:

- Accede al sitio web oficial de MySQL (https://dev.mysql.com/).
- Navega hasta la sección de Descargas.
- Selecciona la versión adecuada para tu sistema operativo (Windows, macOS, Linux).

Descarga de MySQL Workbench:

- Desde la misma página de Descargas en el sitio de MySQL, encuentra y selecciona MySQL Workbench.
- Descarga la versión compatible con tu sistema operativo.

Instalación de MySQL Server:

- Ejecuta el archivo de instalación descargado.
- Sigue las instrucciones del asistente de instalación.
- Durante el proceso, se te solicitará establecer una contraseña para el usuario root de MySQL. Asegúrate de recordar esta contraseña, ya que se utilizará para administrar la base de datos.

Configuración de MySQL Server:

- Al finalizar la instalación, puedes configurar MySQL como un servicio para que se inicie automáticamente con el sistema.
- Selecciona las opciones de configuración según tus preferencias o utiliza la configuración predeterminada.

Inicio del Servicio MySQL:

- Inicia el servicio MySQL.
- Verifica que el servicio esté en ejecución y que no haya problemas de conexión.

Instalación de MySQL Workbench:

- Ejecuta el archivo de instalación de MySQL Workbench descargado.
- Sigue las instrucciones del asistente de instalación.

Conexión a MySQL Server desde Workbench:

- Abre MySQL Workbench.

- Establece una nueva conexión proporcionando la dirección del servidor (generalmente 'localhost'), el nombre de usuario (root por defecto) y la contraseña que estableciste durante la instalación de MySQL Server.

Verificación de la Conexión:

- Después de establecer la conexión, verifica que puedes acceder al servidor MySQL desde MySQL Workbench.

Link de Tutorial para instalar MysqWorkBench y Mysql Sever.

https://www.youtube.com/watch?v=tAxTQoofbeM

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Sistemas Manejadores de Base de Datos.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Crear un modelo de bases de Datos MER.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.

2. Observa el video de la clase magistral.

3. Revisa en el calendario el día y hora en que se debe de realizar el Test.

4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

Tema 2: Sistemas de Gestión de Base de Datos

Objetivo: Analizar los Sistemas de Gestión de Bases de Datos (SGBD), comprendiendo su arquitectura, funcionalidades y tipos, con el propósito de optimizar el almacenamiento, la manipulación y la administración eficiente de los datos en distintos entornos informáticos.

Iniciaremos contextualizando la importancia crítica de los Sistemas de Gestión de Bases de Datos (SGBD) en la actualidad. Resaltaremos cómo estos sistemas se han convertido en pilares fundamentales para la organización y eficiente manejo de grandes volúmenes de datos en diversas industrias.

Imagen 1. Sistemas Gestores de Base de Datos



Fuente: https://basededatos99.weebly.com/base-de-datos/5-tipos-de-manejadores-de-bases-de-datos-que-existen

Definiciones y Componentes de un SGBD

Profundizaremos en las definiciones y abordaremos los componentes esenciales que componen un SGBD. Desde el motor de base de datos que procesa las consultas hasta las interfaces de usuario que permiten a los usuarios interactuar con la base de datos, los estudiantes obtendrán una comprensión detallada de la arquitectura subyacente.

Bases de datos bibliográficas

Solo contienen un subrogante (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 (o de fuentes primarias —ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

Bases de datos dinámicas

Éstas son 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. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de un supermercado, una farmacia, un videoclub o una empresa.

Bases de datos estáticas

Éstas son 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 jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un nodo padre de información puede tener varios hijos. El nodo que no tiene padres es llamado raíz, y a los nodos que no tienen hijos se los conoce como hojas. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

Bases de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP. Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden

ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

Bases de datos transaccionales

Son bases de datos cuyo único fin es el envío y recepción de datos a grandes velocidades, estas bases son muy poco comunes y están dirigidas por lo general al entorno de análisis de calidad, datos de producción e industrial, es importante entender que su fin único es recolectar y recuperar los datos a la mayor velocidad posible, por lo tanto la redundancia y duplicación de información no es un problema como con las demás bases de datos, por lo general para poderlas aprovechar al máximo permiten algún tipo de conectividad a bases de datos relacionales.

Ventajas de los SGBD

Los Sistemas de Gestión de Bases de Datos (SGBD) ofrecen una variedad de ventajas que contribuyen significativamente a la eficiencia y efectividad en la gestión de datos en entornos empresariales. Aquí se presentan algunas de las principales ventajas:

Integridad de Datos: Los SGBD garantizan la integridad de los datos mediante la implementación de reglas y restricciones. Esto evita la entrada de información incorrecta o inconsistente, manteniendo la coherencia de la base de datos.

Seguridad: Los SGBD proporcionan mecanismos de seguridad robustos, como la autenticación y autorización de usuarios, para proteger el acceso no autorizado a la información. Se pueden establecer roles y privilegios para controlar quién puede ver o modificar determinados conjuntos de datos.

Eficiencia en la Recuperación de Datos: Facilitan la recuperación eficiente de datos a través de consultas y búsquedas optimizadas. Los índices y la capacidad de realizar consultas complejas permiten obtener información de manera rápida y precisa.

Consistencia: Los SGBD garantizan la consistencia de los datos al ofrecer transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Esto significa que las operaciones se realizan de manera que la base de datos siempre mantenga un estado consistente, incluso en situaciones de fallo.

Escalabilidad: Permiten la gestión de grandes cantidades de datos y escalabilidad para adaptarse al crecimiento de la empresa. Los SGBD pueden manejar

eficientemente desde pequeñas bases de datos locales hasta sistemas distribuidos a gran escala.

Mantenimiento Facilitado: Los cambios en la estructura de la base de datos o en las reglas de negocio se pueden realizar de manera controlada y sistemática, minimizando el impacto en las aplicaciones que utilizan la base de datos.

Reducción de la Redundancia de Datos: Los SGBD facilitan la normalización de la base de datos, eliminando la redundancia y mejorando la eficiencia del almacenamiento de datos. Esto ayuda a mantener la coherencia y evita inconsistencias.

Facilidad en el Acceso Concurrente: Soportan múltiples usuarios accediendo y modificando datos de manera concurrente, gestionando eficientemente las solicitudes y asegurando que las transacciones se completen de manera coherente.

Facilidad en la Realización de Copias de Seguridad y Restauración: Los SGBD proporcionan herramientas integradas para realizar copias de seguridad y restaurar datos. Esto asegura la disponibilidad y recuperación rápida de la información en caso de fallos o pérdidas.

Cumplimiento de Normativas y Estándares: Algunos SGBD están diseñados para cumplir con normativas y estándares específicos de la industria, lo que es esencial en sectores regulados como la salud o las finanzas.

Imagen 2. Ventajas Sistemas Gestores de Base de Datos



Fuente: https://basededatos99.weebly.com/base-de-datos/5-tipos-de-manejadores-de-bases-de-datos-que-existen

“SISTEMAS MANEJADORES DE BASE DE DATOS”

CUADRO COMPARATIVO

MANEJADOR DE BASE DE DATOS	PRECIO	SEGURIDAD	FLEXIBILIDAD	ESTABILIDAD	CAPACIDAD	VENTAJAS	DESVENTAJAS	REQUERIMIENTOS HARWARD	MI SISTEMA
	**Versión Precio** SQL Server 2008 Enterprise Edition $24,999.00 SQL Server 2008 Standard Edition $5,999.00	cuenta con un esquema de seguridad muy interesante tanto para la validación del usuario como la seguridad		Hace más fácil para las organizaciones cumplir las políticas de cumplimiento.	SQL Server admite 25 instancias en un clúster de conmutación por error cuando se usa un disco de clúster compartido como opción de almacenami	❖ Administración multiservidor y con una sola consola. ❖ Ejecución y alerta de trabajos basadas en eventos. ❖ Seguridad integrada.	❖ Gran cantidad de memoria RAM ara la instalación y utilización del software. ❖ Costo de actualizaciones. ❖ La relación calidad-precio está muy debajo	RAM: Mínimo: 512 MB Recomendado: 2,048 GB o más	✓
								Tipo de procesador: Procesador Itanium o más rápido	×

	SQL Server 2008 Workgroup Edition $3,899.00 SQL Server 2008 Developer Edition $49.95	de los datos			ento para la instalación del clúster; SQL Server admite 50 instancias en un clúster de conmutación por error si elige recursos compartidos de archivos SMB como opción de almacenamiento para la instalación del clúster.	❖ Se ajusta muy bien a las necesidades cada vez mayores. ❖ Para las bases de datos muy grandes. ❖ Ideal para sistemas de alta tecnología.	comparado con Oracle.	Velocidad de procesador: Recomendado: 1,0 GHz o más	
	80 y 400 dólares dependien	Oracle pone al alcance	La tecnología Oracle para organizaciones	Es muy resistente.	La capacidad de BDD es	❖ Suite de productos que ofrece una	❖ Sucedieron varias versiones con	RAM :256MB mínimo	✓

ORACLE	do del tipo de licencia de usuario	del DBA varios niveles de seguridad: - Seguridad de cuentas para la validación de usuario. - Seguridad en el acceso a los objetos de la base de datos. - Seguridad	en crecimiento es abierta, fácil de integrar, simple de administrar y altamente escalable, lo cual ayuda a adaptarse más rápido a los cambios del mercado o a los requerimientos del cliente. Las soluciones Oracle permiten alinear sus procesos con las cambiantes necesidades durante un		alta ya que soporta hasta 4 peta bytes de información.	gran variedad de herramientas. ❖ Oracle corre en computadoras personales, microcomputadoras, mainframes y computadoras con procesamientos paralelo masivo. ❖ Soporta unos 17 idiomas. ❖ Corre automáticamente en mas de 80 arquitecturas	correcciones, hasta alcanzar la estabilidad en la 8.0.3. ❖ Oracle mal configurado puede ser desesperantemente lento		
								MEMORIA VIRTUAL: doble de la cantidad de RAM	✓
								Espacio de disco duro: se especificará a continuación Espacio para archivos temporales :100 Mb	✓
								Adaptadores de video :256 colores	✓

		a nivel de sistema para la gestión de privilegios globales.	largo período, brindando así valor continuo para su empresa.			de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. ❖ Oracle es la base de datos con más orientación hacia internet.		Procesador: 200 MHz mínimo Protocolo de red TCP/IP	 ✓
	Sin costo.	Para encontrar información específica sobre el sistema de control	Es sumamente fácil de administrar, operar y es famoso por su instalación de 15 minutos, la cual en 15	Intentaremos clarificar estas cuestiones y responder algunas preguntas importantes	Win32 w/ FAT/FAT32 2GB/4GB Win32 w/ NTFS 2TB Linux 2.2-Intel 32-bit	❖ Buen rendimiento, buena velocidad a la hora de conectar con el servidor y	❖ No soporta vistas (entre otras cosas).	RAM :512 MB Memoria virtual[1]: 1024 MB	✓ ✓

		de accesos que MySQL. Utiliza para crear cuentas de usuarios y comproba r el acceso a las bases de datos	minutos es posible instalar, configurar y montar una BD relacional ya sea sistemas OLTP o bien OLAP, aunque por excelencia MySQL Server es comúnmente usado para desarrollos We b 2.0 y Enterpris e2.0.	que preocupan a muchos usuarios potenciales . La información en esta sección se basa en datos recopilados de las listas de correo, que son muy activas para identificar problemas, así como para reportar tipos de usos.	2GB (LFS: 4GB) Linux 2.4+ (usando ext3 filesystem) 4TB Solaris 9/10 16TB MacOS X w/ HFS+ 2TB NetWare w/NSS filesystem 8TB	de respuesta a consultas. ❖ Registros sin límite de tamaño. ❖ Control de acceso: qué usuarios tienen acceso a qué tablas y con qué permisos.		Protocolo de red TCP/IP	✓
								Protocolo de red TCP/IP con SSL	×

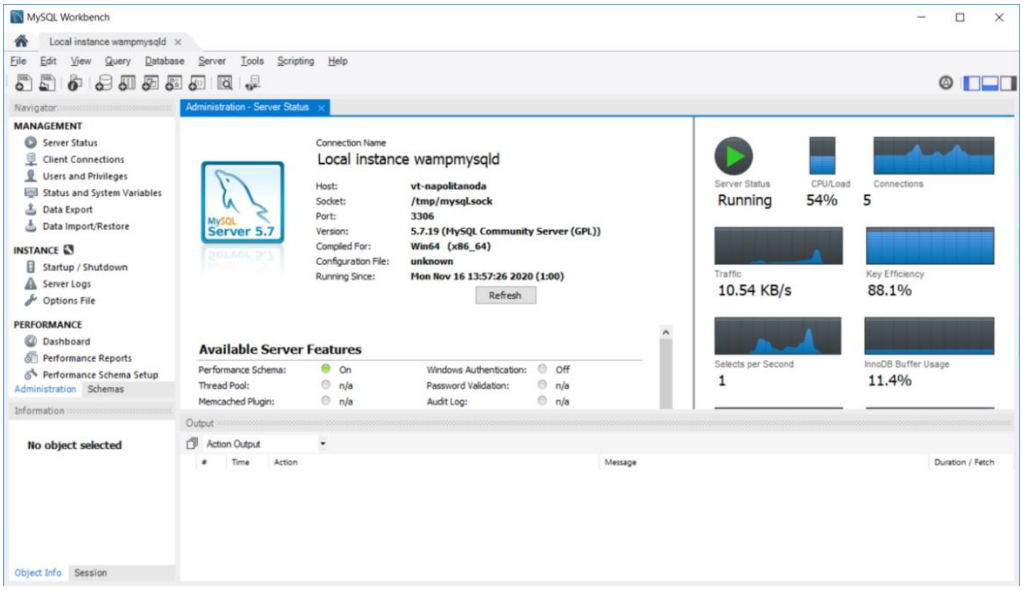
Configuración de SGBD

Configurar un Sistema de Gestión de Bases de Datos (SGBD) es un proceso crucial para asegurar su correcto funcionamiento y adaptarlo a las necesidades específicas de un entorno particular. Aquí se presentan los pasos generales para configurar un SGBD:

Elección del SGBD:

Selecciona el SGBD que mejor se adapte a los requisitos del proyecto. Ejemplos comunes incluyen MySQL, PostgreSQL, Microsoft SQL Server, Oracle, MongoDB, entre otros.

Imagen 3. Sistemas Gestores de Base de Datos / Workbench



Fuente: https://basededatos99.weebly.com/base-de-datos/5-tipos-de-manejadores-de-bases-de-datos-que-existen

Descarga e Instalación:

Descarga la versión más reciente del SGBD desde el sitio web oficial del proveedor.

Sigue las instrucciones de instalación proporcionadas por el proveedor. Este paso puede incluir la configuración de rutas de instalación, opciones de idioma y configuraciones predeterminadas.

Configuración Inicial:

Después de la instalación, realiza la configuración inicial del SGBD. Este proceso puede incluir la especificación de la ubicación de los archivos de datos y registros, así como la configuración de parámetros globales como el tamaño de la memoria y la cantidad de conexiones permitidas.

Creación de Usuarios y Privilegios:

Crea cuentas de usuario para acceder a la base de datos. Es recomendable asignar privilegios específicos a cada usuario para garantizar la seguridad y el principio de privilegios mínimos necesarios.

Establecimiento de Contraseñas Seguras:

Implementa políticas de contraseñas seguras para los usuarios. Esto incluye la especificación de requisitos como longitud mínima, el uso de caracteres especiales y la caducidad de contraseñas.

Configuración de Red:

Configura la red para permitir conexiones remotas si es necesario. Define las configuraciones de red para permitir que otras aplicaciones o servidores se conecten al SGBD.

Configuración de Copias de Seguridad y Restauración:

Establece políticas y programaciones para la realización de copias de seguridad periódicas. Asegúrate de comprender cómo restaurar datos desde estas copias de seguridad en caso de pérdida o corrupción.

Ajuste de Configuraciones Avanzadas:

Realiza ajustes avanzados según las necesidades del sistema. Esto puede incluir la configuración de índices, particiones, políticas de almacenamiento y otras configuraciones específicas del SGBD.

Pruebas de Conectividad y Rendimiento:

Realiza pruebas de conectividad para garantizar que los usuarios y aplicaciones puedan acceder al SGBD correctamente. También realiza pruebas de rendimiento para evaluar y ajustar la configuración según sea necesario.

Documentación:

Documenta la configuración realizada, incluyendo detalles como las configuraciones de seguridad, usuarios y privilegios, y cualquier ajuste específico del sistema. Esto será útil para referencia futura y para compartir conocimientos con otros administradores.

Estos pasos proporcionan una guía general para la configuración de un SGBD. Es importante consultar la documentación específica del SGBD elegido para obtener detalles exactos sobre cada paso y considerar las mejores prácticas recomendadas por el proveedor.

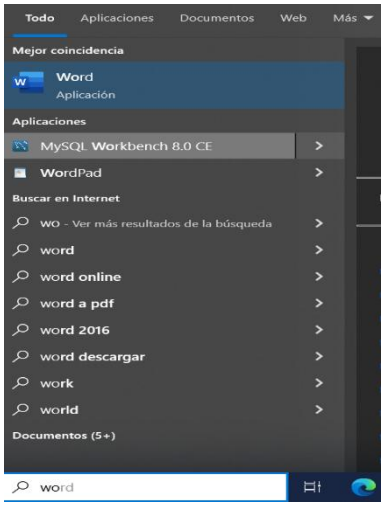
Creación de Conexiones a Base de Datos

Configurar conexiones en MySQL Workbench es esencial para interactuar con bases de datos. Aquí te dejo los pasos generales para crear conexiones en MySQL Workbench:

Inicio de MySQL Workbench:

Inicia MySQL Workbench desde tu sistema. Puedes encontrarlo en el menú de programas o aplicaciones, según tu sistema operativo.

Imagen 4. Inicio Rápido / Workbench



Fuente: elaboración propia

Pestaña "Inicio de sesión/Inicio Rápido":

En la pantalla principal, busca la pestaña "Inicio de sesión" o "Inicio Rápido". Este suele ser el lugar donde puedes comenzar a configurar nuevas conexiones.

Añadir Nueva Conexión:

Haz clic en la opción que te permite "Añadir nueva conexión" o un botón similar.

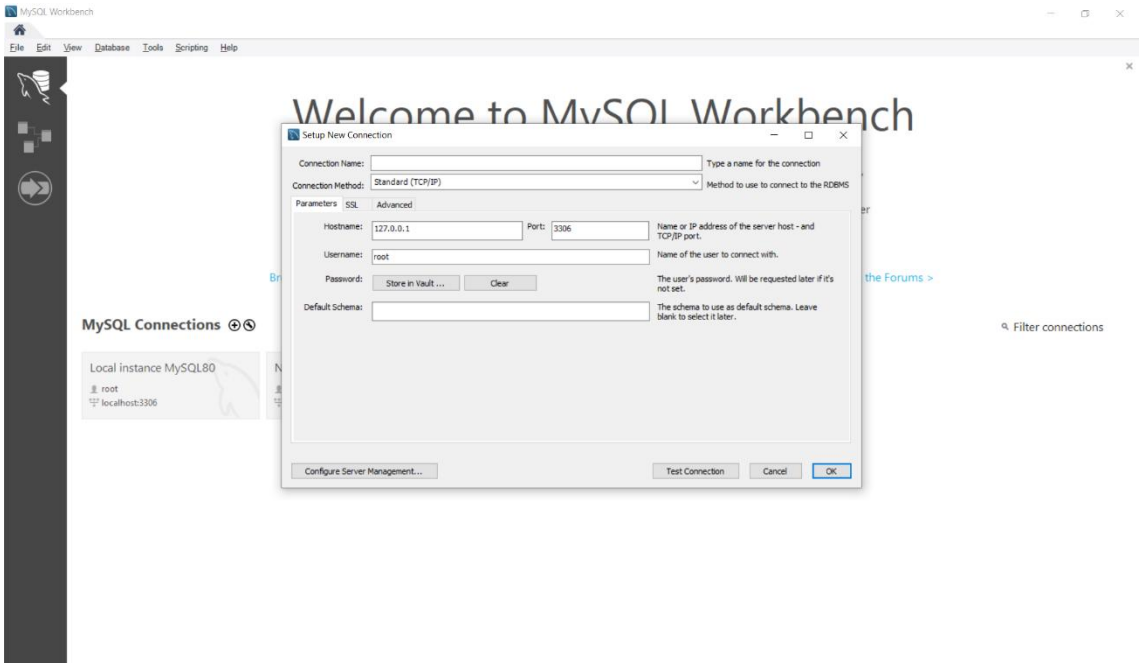


Imagen 5. Pantalla de Inicio / Workbench

Fuente: elaboración propia

Configuración de Conexión:

Aparecerá una ventana que te permite ingresar los detalles de la conexión.

Nombre de la Conexión: Asigna un nombre descriptivo para la conexión.

Nombre/IP del Host: Indica la dirección IP o el nombre del servidor donde se aloja la base de datos.

Puerto: Especifica el número de puerto que utiliza el servidor de base de datos (el valor predeterminado para MySQL es 3306).

Nombre de Usuario y Contraseña: Proporciona las credenciales de acceso al servidor de la base de datos.

Base de Datos Predeterminada: Selecciona la base de datos a la que deseas conectarte por defecto.

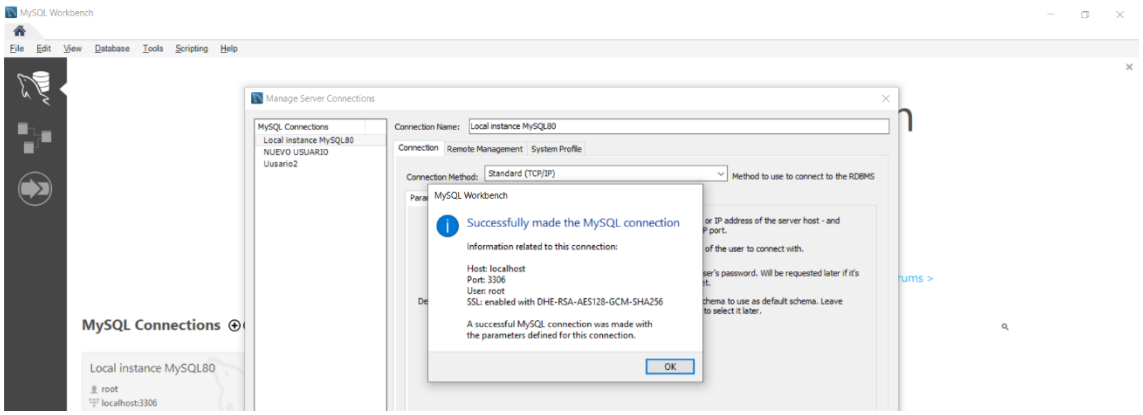
Imagen 5. Crear Conexión / Workbench

Fuente: elaboración propia

Probar Conexión:

Algunas configuraciones de MySQL Workbench permiten probar la conexión antes de guardarla. Utiliza esta opción para asegurarte de que los detalles ingresados sean correctos.

Imagen 6. Probar Conexión / Workbench



Fuente: elaboración propia

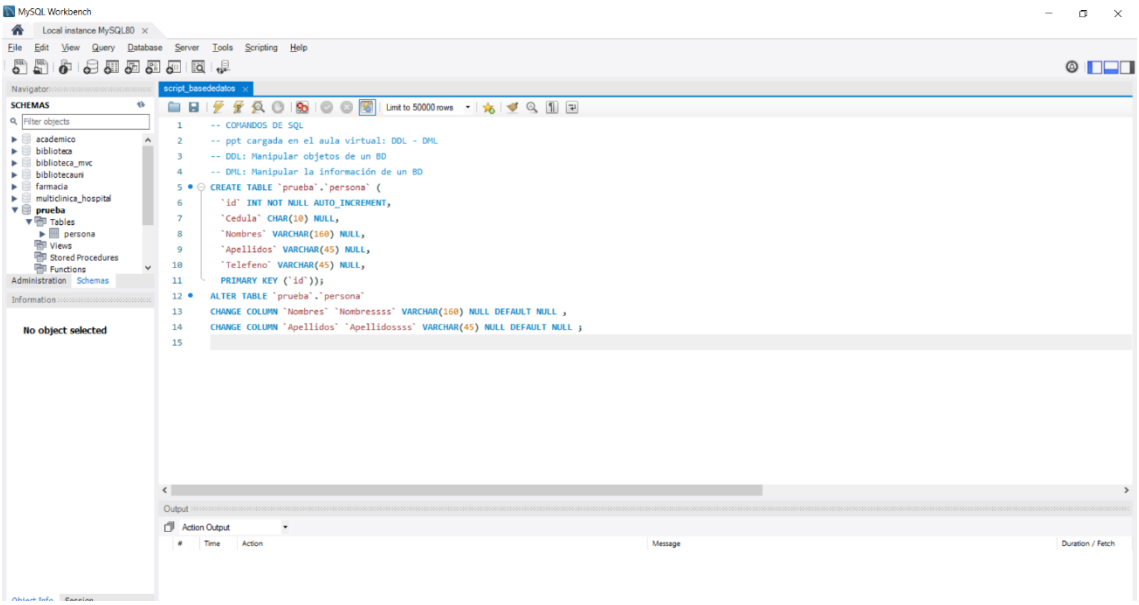
Guardar Conexión:

Si la prueba es exitosa, guarda la conexión. Este paso podría implicar hacer clic en un botón como "Aceptar" o "Guardar". Asegúrate de que la conexión aparezca en la lista de conexiones disponibles.

Conectar a la Base de Datos:

Después de guardar la conexión, puedes seleccionarla de la lista y hacer clic en "Conectar" para establecer la conexión con la base de datos.

Imagen 7. Conectar a Base de Datos / Workbench



Fuente: elaboración propia

Explorar y Trabajar con la Base de Datos:

Una vez conectado, puedes explorar la estructura de la base de datos, ejecutar consultas SQL y realizar diversas operaciones de gestión de datos.

Estos pasos proporcionan una guía general para configurar conexiones en MySQL Workbench. Ten en cuenta que los detalles exactos pueden variar según la versión específica de MySQL Workbench que estés utilizando.

Consulta la documentación de MySQL Workbench para obtener instrucciones detalladas y actualizadas.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: WorkBench de MySQL.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Crear una base de datos de nombre Prueba con una tabla que se llame persona.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.
2. Observa el video de la clase magistral.
3. Revisa en el calendario el día y hora en que se debe de realizar el Test.
4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

III. DATOS GENERALES DE LA UNIDAD 3

Información General

1.1.	Nivel	Segundo					
1.2.	Nombre Unidad	Introducción a sistemas de bases de datos.					
1.3.	Tema 1 – Unidad	Introducción a los conceptos de base de datos.					
1.4.	Tema 2 – Unidad	Modelado de Datos Lógicos de Base de datos.					
1.5.	Unidad organizacional	Arquitectura de Sistemas de Base de datos					
1.6.	Unidad	3					
1.7.	Total, Horas Unidad	15					
1.8.	Detalle de horas Unidad	ACD	6	AA	4	APE	5

Resultado de Aprendizaje

Al finalizar esta unidad, los estudiantes serán capaces de recuperar, insertar, actualizar y eliminar información de una base de datos utilizando Transact/SQL y aplicando múltiples condiciones de comparación o funciones para el manejo de campos tipo fecha. construir programas estructurados y manejar los errores utilizando el lenguaje Transact-SQL.

Metodología

La Metodología de se basa en un enfoque activo y colaborativo. Incluye sesiones de clase estructuradas en torno a temas específicos, lecturas y tareas, estudio de casos, proyectos prácticos y debates en clase. Los estudiantes también se involucran en trabajos en grupo y mantienen diarios de reflexión. Se fomenta la participación en debates y preguntas y respuestas, y se proporcionan recursos en línea para una comprensión más profunda. La evaluación se realiza mediante pruebas, proyectos y una evaluación final, con retroalimentación continua para guiar el aprendizaje. Esta metodología busca que los estudiantes comprendan los conceptos fundamentales y puedan aplicar eficazmente las metodologías de desarrollo de software en situaciones prácticas.

DESARROLLO

Unidad 3: Estructuras internas de almacenamiento y estructura de índices

Tema 1: Arquitectura del sistema de base de datos

Objetivo: Analizar la arquitectura de los sistemas de bases de datos, comprendiendo sus niveles, componentes y modelos de almacenamiento, con el propósito de optimizar la gestión, el acceso y la seguridad de la información en entornos computacionales.

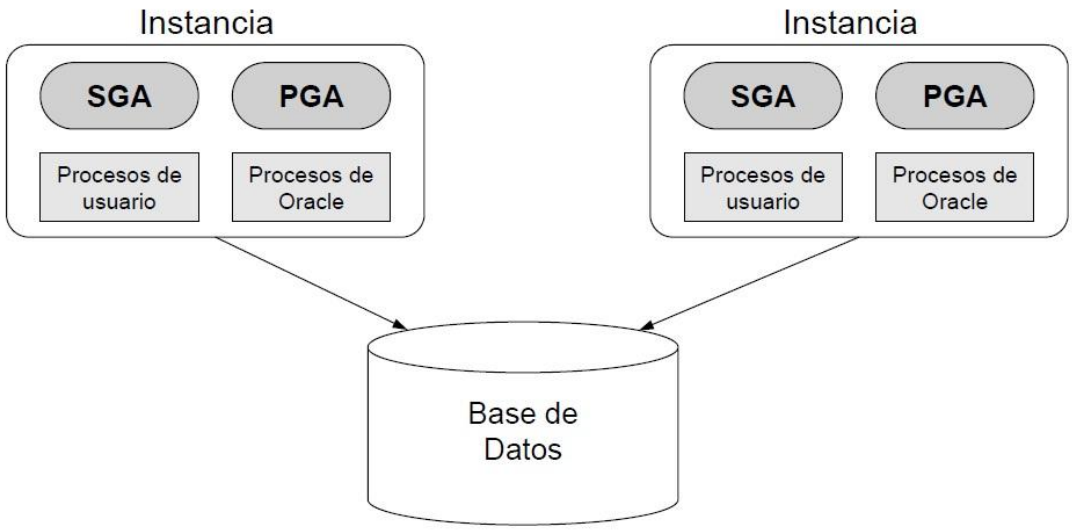
Instancia de Sistemas de Base de Datos

La instancia de un Sistema de Gestión de Bases de Datos (SGBD) se refiere a un conjunto específico de datos almacenados en una base de datos en un momento particular. Aquí te proporcionaré información detallada sobre este concepto, incluyendo definiciones, ejemplos, comparaciones y referencias:

Definición:

Una instancia de base de datos se puede definir como la representación concreta de la base de datos en un momento específico. Es el contenido actual y observable de la base de datos en términos de registros, tablas y relaciones entre los datos.

Imagen 1: Instancia de una base de datos



Fuente: https://dan1456bd.blogspot.com/p/conjunto-de-estructuras-de-memoria-y.html

Ejemplo:

Supongamos que tienes una base de datos llamada "Empresa" que almacena información sobre empleados. La instancia de esta base de datos en un momento particular podría ser la lista actual de empleados, sus roles, salarios y otra información específica en ese momento.

Comparación con Modelo de Datos:

Mientras que el modelo de datos describe la estructura y el diseño de la base de datos, la instancia es una instantánea de cómo se ven realmente los datos en un momento dado. La estructura (modelo) define las tablas, relaciones y restricciones, mientras que la instancia es la representación real de los datos basada en ese modelo.

Importancia en la Administración de Bases de Datos:

La capacidad de comprender y gestionar instancias de bases de datos es esencial para los administradores de bases de datos. Permite realizar tareas como realizar copias de seguridad, realizar ajustes de rendimiento y garantizar la consistencia de los datos en diferentes momentos.

En resumen, la instancia de un sistema de base de datos representa el estado actual y concreto de los datos en un momento específico, y comprender este concepto es fundamental para trabajar eficientemente con bases de datos en entornos reales.

Comandos DDL en Base de datos

Los comandos DDL son un conjunto de instrucciones utilizadas para definir, modificar y eliminar la estructura de una base de datos. Aquí te proporcionaré información detallada sobre los Comandos DDL, incluyendo definiciones, ejemplos, comparaciones y referencias:

Definición:

Los Comandos DDL (Lenguaje de Definición de Datos) son un conjunto de instrucciones SQL que permiten definir y gestionar la estructura de la base de datos. Estos comandos se utilizan para crear, modificar y eliminar objetos de la base de datos, como tablas, índices, vistas y restricciones.

Imagen 2: Comandos DDL



Fuente: https://es.slideshare.net/MirianJanethPaquirac/comandos-ddl-93265007

Principales Comandos DDL:

CREATE:

Utilizado para crear nuevos objetos en la base de datos, como tablas, índices o vistas.

Ejemplo: CREATE TABLE Empleados (ID INT, Nombre VARCHAR(50), Salario FLOAT);

ALTER:

Modifica la estructura existente de la base de datos, como agregar o eliminar columnas.

Ejemplo: ALTER TABLE Empleados ADD COLUMN Departamento VARCHAR(30);

DROP:

Elimina objetos de la base de datos, como tablas o índices.

Ejemplo: DROP TABLE Empleados;

TRUNCATE:

Elimina todos los registros de una tabla, pero conserva la estructura de la tabla.

Ejemplo: TRUNCATE TABLE Empleados;

COMMENT:

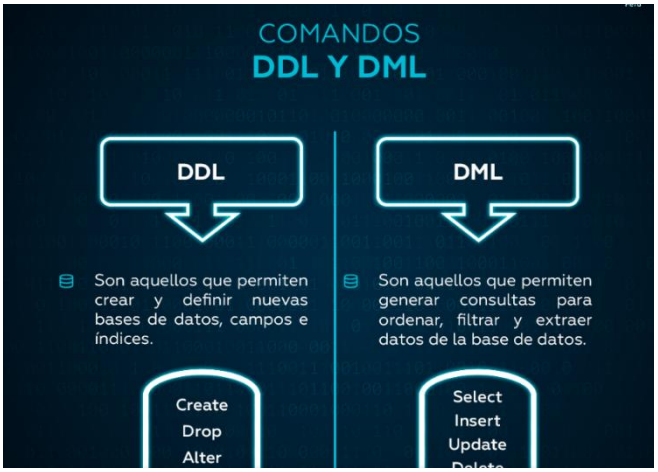
Permite agregar comentarios o descripciones a objetos de la base de datos.

Ejemplo: COMMENT ON COLUMN Empleados.Nombre IS 'Nombre del empleado';

Comparación con DML (Data Manipulation Language):

Mientras que los comandos DDL se centran en la estructura de la base de datos, los comandos DML se utilizan para manipular y gestionar los datos almacenados en la base de datos. DDL se enfoca en "qué" estructuras existen, mientras que DML se centra en "cómo" interactuamos con los datos.

Imagen 3: Comandos DDL – DML - Comparativa



Fuente: https://es.slideshare.net/MirianJanethPaquirac/comandos-ddl-93265007

Ejemplo Práctico:

Supongamos que queremos crear una nueva tabla de productos en una base de datos de comercio electrónico:

```
CREATE TABLE Productos (
  ProductoID INT PRIMARY KEY,
  Nombre VARCHAR(100),
  Precio FLOAT,
  Stock INT
);
```

Este comando DDL crea una nueva tabla llamada "Productos" con columnas para identificación del producto, nombre, precio y stock.

Los Comandos DDL son esenciales para la administración y diseño efectivo de bases de datos, permitiendo a los desarrolladores y administradores definir la estructura que almacena y organiza los datos de manera eficiente.

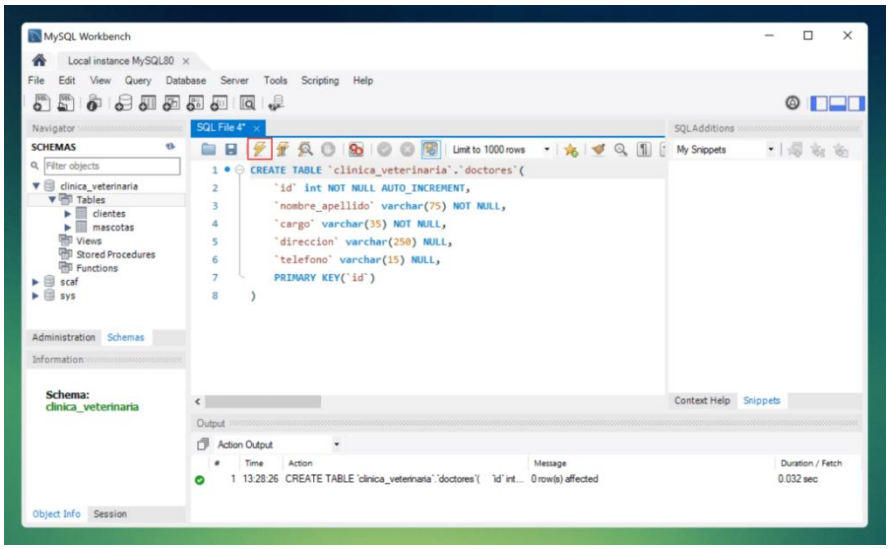
Tablas, llaves primarias y secundarias en una base de datos

En una base de datos relacional, las tablas son estructuras fundamentales que almacenan datos organizados en filas y columnas. Las llaves primarias y secundarias son elementos clave para establecer relaciones entre tablas.

Tablas:

Definición: Una tabla es una estructura que organiza los datos en filas y columnas. Cada fila representa una entrada única y cada columna almacena un atributo específico de esos datos.

Imagen 4: Tablas en Workbench



Fuente: https://norvicsoftware.com/crear-tablas-en-mysql/

Ejemplo:

```
CREATE TABLE Empleados (
  ID INT PRIMARY KEY,
  Nombre VARCHAR(50),
  Departamento VARCHAR(30),
```

```
  PRIMARY KEY (ID)
);
```

En este ejemplo, la tabla "Empleados" tiene columnas para ID, Nombre y Departamento.

Llave Primaria (Primary Key):

Definición: La llave primaria es un campo o conjunto de campos que identifica de manera única cada fila en una tabla. Garantiza que no puede haber duplicados y que cada registro puede ser identificado de manera única.

Ejemplo:

```
CREATE TABLE Empleados (
  ID INT PRIMARY KEY,
  Nombre VARCHAR(50),
  Departamento VARCHAR(30)
);
```

En este caso, "ID" es la llave primaria.

Llave Secundaria (Foreign Key):

Definición: Una llave secundaria establece una relación entre dos tablas. Es un campo que apunta a la llave primaria en otra tabla y se utiliza para vincular registros entre tablas.

Ejemplo:

```
CREATE TABLE Departamentos (
  ID INT PRIMARY KEY,
  Nombre VARCHAR(30)
);

CREATE TABLE Empleados (
```

```
  ID INT PRIMARY KEY,
  Nombre VARCHAR(50),
  DepartamentoID INT,
  FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(ID)
);
```

Aquí, "DepartamentoID" en la tabla "Empleados" es una llave secundaria que referencia la llave primaria "ID" en la tabla "Departamentos".

Comparación:

La llave primaria identifica de manera única registros en su propia tabla, mientras que la llave secundaria establece relaciones entre tablas.

La llave primaria se define en la creación de la tabla, mientras que la llave secundaria se agrega con la declaración FOREIGN KEY.

Entender el diseño de tablas y el uso adecuado de llaves primarias y secundarias es crucial para la creación de bases de datos relacionales efectivas y eficientes. Estos elementos forman la base para establecer relaciones y garantizar la integridad de los datos en un entorno de bases de datos.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Comandos DDL, ¿consideras aplicarlos manualmente o con ayuda del asistente?

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Identificar opciones del SMBD para crear y manipular objetos de una base de datos.

Crear base de datos y tablas según caso de estudio.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.
2. Observa el video de la clase magistral.
3. Revisa en el calendario el día y hora en que se debe de realizar el Test.
4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

Tema 2: Objetos en Bases de Datos

Objetivo: Explorar los objetos en bases de datos, comprendiendo su estructura, tipos y funciones, con el propósito de optimizar la organización, manipulación y gestión eficiente de la información dentro de los sistemas de bases de datos.

En el contexto de bases de datos, los "objetos" son elementos fundamentales que componen la estructura y funcionalidad de una base de datos. Aquí te proporcionaré información detallada, ejemplos, comparaciones y referencias sobre los objetos de una base de datos:

Definición:

Los objetos de una base de datos son entidades que permiten organizar y gestionar datos de manera estructurada. Estos objetos pueden variar según el sistema de gestión de bases de datos (SGBD), pero algunos comunes incluyen tablas, vistas, índices, procedimientos almacenados y desencadenadores.

Principales Objetos de una Base de Datos:

Tablas: Las tablas son objetos principales que almacenan datos en filas y columnas.

Ejemplo:

```
CREATE TABLE Empleados (
  ID INT PRIMARY KEY,
  Nombre VARCHAR(50),
  Salario FLOAT
);
```

Vistas: Las vistas son representaciones virtuales de los datos almacenados en una o más tablas.

Ejemplo:

```
CREATE VIEW VistaEmpleados AS

SELECT ID, Nombre FROM Empleados WHERE Salario > 50000;
```

Índices Los índices mejoran la velocidad de recuperación de datos al proporcionar una referencia rápida a los registros.

Ejemplo:

```
CREATE INDEX idx_Salario ON Empleados (Salario);
```

Procedimientos Almacenados: Los procedimientos almacenados son conjuntos de instrucciones SQL que se almacenan y pueden ser ejecutados repetidamente.

Ejemplo:

```
CREATE PROCEDURE ObtenerEmpleados

AS

SELECT * FROM Empleados;
```

Desencadenadores (Triggers): Los desencadenadores son instrucciones que se ejecutan automáticamente en respuesta a ciertos eventos en la base de datos.

Ejemplo:

```
CREATE TRIGGER ActualizarSalario

AFTER UPDATE ON Empleados

FOR EACH ROW

SET NEW.Salario = NEW.Salario * 1.1;
```

Comparación de Objetos:

Las tablas almacenan los datos, las vistas proporcionan vistas personalizadas de esos datos, los índices mejoran el rendimiento de las consultas, los procedimientos almacenados contienen lógica de negocio y los desencadenadores responden a eventos específicos.

Procedimientos almacenados en una base de datos

En una base de datos, los procedimientos almacenados son un tipo de objeto que contiene una o más instrucciones SQL y se almacena en el sistema de gestión de bases de datos (SGBD) para su posterior ejecución. A continuación, te proporcionaré información detallada, ejemplos, comparaciones y referencias sobre los procedimientos almacenados:

Un procedimiento almacenado es un conjunto de instrucciones SQL predefinidas que se guardan en la base de datos y se pueden ejecutar repetidamente. Estos procedimientos pueden aceptar parámetros y contener lógica de negocio, proporcionando modularidad y reutilización de código.

Ejemplo Básico:

Supongamos que queremos crear un procedimiento almacenado que devuelve la lista de empleados de un departamento específico:

```
CREATE PROCEDURE ObtenerEmpleadosPorDepartamento(IN DepartamentoID INT)

BEGIN

  SELECT * FROM Empleados WHERE DepartamentoID = DepartamentoID;

END;
```

En este ejemplo, el procedimiento ObtenerEmpleadosPorDepartamento toma un parámetro DepartamentoID y devuelve todos los empleados asociados a ese departamento.

Ventajas de Procedimientos Almacenados:

Reutilización de Código:

Los procedimientos almacenados permiten escribir código SQL complejo una vez y reutilizarlo en múltiples partes de una aplicación.

Modularidad:

Al dividir la lógica de negocio en procedimientos almacenados, se facilita el mantenimiento y la actualización del sistema.

Seguridad:

Los procedimientos almacenados pueden controlar el acceso a los datos, ya que los usuarios pueden tener permisos para ejecutar el procedimiento, pero no para acceder directamente a las tablas.

Optimización de Rendimiento:

Al ejecutar un procedimiento almacenado, el plan de ejecución puede almacenarse en caché, mejorando la eficiencia y el rendimiento de las consultas.

Comparación con Consultas Ad Hoc:

Mientras que las consultas ad hoc son instrucciones SQL que se escriben y ejecutan de forma independiente en el momento, los procedimientos almacenados ofrecen una encapsulación de la lógica de negocio que puede ser invocada cuando sea necesario.

Ejemplo Avanzado:

Supongamos que queremos incrementar el salario de todos los empleados en un departamento específico en un cierto porcentaje. Podemos crear un procedimiento almacenado para esto:

```
CREATE PROCEDURE IncrementarSalariosPorDepartamento(IN DepartamentoID INT, IN Porcentaje FLOAT)

BEGIN

  UPDATE Empleados SET Salario = Salario * (1 + Porcentaje/100) WHERE DepartamentoID = DepartamentoID;

END;
```

Este procedimiento IncrementarSalariosPorDepartamento toma dos parámetros: DepartamentoID y Porcentaje, y actualiza los salarios en consecuencia.

Los procedimientos almacenados son herramientas poderosas en el diseño de bases de datos, proporcionando una capa de abstracción que mejora la seguridad, el rendimiento y la mantenibilidad de las aplicaciones que interactúan con la base de datos.

Funciones en una Base de datos

En el contexto de bases de datos, las funciones son herramientas que realizan cálculos o manipulan datos para devolver un resultado. A continuación, te proporcionaré información detallada, ejemplos, comparaciones y referencias sobre funciones en base de datos:

Definición:

Las funciones en bases de datos son bloques de código que aceptan parámetros, realizan operaciones específicas en los datos y devuelven un valor. Pueden utilizarse para realizar cálculos, manipular cadenas, fechas o realizar otras operaciones en los datos almacenados en la base de datos.

Tipos de Funciones:

Funciones de Agregación:

Definición: Realizan operaciones en conjuntos de datos y devuelven un único valor. Ejemplos incluyen SUM, AVG, COUNT, MAX y MIN.

Ejemplo:

SELECT AVG(Salario) FROM Empleados;

Funciones Escalares:

Definición: Operan en cada fila de un conjunto de datos y devuelven un valor para cada fila.

Ejemplo:

SELECT Nombre, UPPER(Nombre) AS NombreMayusculas FROM Empleados;

Funciones de Cadena:

Definición: Manipulan cadenas de texto. Ejemplos incluyen CONCAT, LENGTH, SUBSTRING, UPPER, LOWER.

Ejemplo:

SELECT CONCAT(FirstName, ' ', LastName) AS NombreCompleto FROM Personas;

Funciones de Fecha y Hora:

Definición: Operan en datos de fecha y hora. Ejemplos incluyen NOW, DATE, EXTRACT, DATE_ADD.

Ejemplo:

SELECT * FROM Pedidos WHERE FechaPedido > DATE_ADD(NOW(), INTERVAL -7 DAY);

Comparación con Procedimientos Almacenados:

Aunque tanto funciones como procedimientos almacenados contienen lógica de negocio, las funciones devuelven un valor y se utilizan en expresiones SQL, mientras que los procedimientos almacenados pueden ejecutar múltiples instrucciones y no necesariamente devuelven un valor.

Ventajas de Funciones:

Reutilización de Código:

Las funciones permiten escribir lógica de negocio una vez y reutilizarla en diferentes partes de las consultas SQL.

Simplicidad y Claridad:

El uso de funciones puede simplificar y hacer más legible el código SQL al encapsular operaciones complejas.

Optimización de Rendimiento:

Las funciones pueden ser optimizadas por el sistema de gestión de bases de datos para mejorar el rendimiento de las consultas.

Ejemplo Avanzado:

Supongamos que queremos calcular la edad promedio de los empleados en la base de datos:

```
SELECT AVG(YEAR(NOW()) - YEAR(FechaNacimiento)) AS EdadPromedio FROM Empleados;
```

En este ejemplo, la función YEAR se utiliza para extraer el año de la fecha actual y la fecha de nacimiento de cada empleado, calculando así la edad.

Las funciones en bases de datos son herramientas esenciales para realizar cálculos y manipulaciones de datos, mejorando la flexibilidad y eficiencia en el manejo de información en entornos de bases de datos.

Disparadores y Paquetes en una Base de Datos

En el entorno de las bases de datos, los disparadores (triggers) y los paquetes son herramientas avanzadas que permiten automatizar y controlar acciones dentro de la base de datos. A continuación, te proporcionaré información detallada, ejemplos, comparaciones y referencias sobre disparadores y paquetes:

Disparadores (Triggers): Los disparadores son procedimientos almacenados que se activan automáticamente en respuesta a ciertos eventos que ocurren en la base de datos, como la inserción, actualización o eliminación de datos en una tabla.

Paquetes: Los paquetes son objetos de base de datos que contienen un conjunto de procedimientos, funciones, variables y tipos de datos relacionados que se utilizan para realizar una tarea específica en la base de datos.

Ejemplo de Disparador:

Supongamos que queremos registrar cada vez que se inserta un nuevo empleado en la tabla "Empleados". Podemos crear un disparador para esto:

```
CREATE TRIGGER RegistroNuevoEmpleado

AFTER INSERT ON Empleados

FOR EACH ROW

BEGIN

    INSERT INTO RegistroEmpleados (EmpleadoID, FechaRegistro)

    VALUES (NEW.ID, NOW());
```

END;

En este ejemplo, el disparador "RegistroNuevoEmpleado" se activa automáticamente después de que se inserta una nueva fila en la tabla "Empleados", registrando el ID del empleado y la fecha de registro en la tabla "RegistroEmpleados".

Ejemplo de Paquete:

Supongamos que queremos encapsular un conjunto de funciones relacionadas con la gestión de empleados en un paquete llamado "GestionEmpleados":

```
CREATE PACKAGE GestionEmpleados AS

  PROCEDURE AumentarSalario(EmpleadoID INT, Porcentaje FLOAT);

  PROCEDURE DesactivarEmpleado(EmpleadoID INT);

END GestionEmpleados;
```

En este ejemplo, el paquete "GestionEmpleados" contiene dos procedimientos: "AumentarSalario" y "DesactivarEmpleado", que pueden ser invocados desde cualquier parte de la base de datos.

Comparación:

Disparadores vs. Paquetes: Los disparadores se activan en respuesta a eventos específicos de la base de datos, mientras que los paquetes son conjuntos de procedimientos y funciones que pueden ser llamados desde cualquier lugar de la base de datos.

Ventajas y Aplicaciones:

Automatización de Tareas: Tanto los disparadores como los paquetes permiten automatizar tareas y procesos en la base de datos, reduciendo la necesidad de intervención manual.

Integridad de los Datos: Los disparadores pueden utilizarse para garantizar la integridad de los datos, aplicando reglas de negocio y restricciones en la base de datos.

Reutilización de Código: Los paquetes proporcionan un mecanismo para encapsular y reutilizar código relacionado en la base de datos, promoviendo el modularidad y el mantenimiento del código.

Los disparadores y paquetes son herramientas poderosas en la gestión y automatización de tareas en bases de datos, permitiendo un mayor control y eficiencia en el manejo de datos y la lógica de negocio. Su comprensión y aplicación adecuadas son fundamentales para el desarrollo y mantenimiento de sistemas de bases de datos robustos.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Comandos DDL, Diferencias entre Procedimiento almacenado y vistas.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Crear script con comandos DDL y DML, según caso de estudio.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.
2. Observa el video de la clase magistral.
3. Revisa en el calendario el día y hora en que se debe de realizar el Test.
4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

IV. DATOS GENERALES DE LA UNIDAD 4

Información General

1.1.	Nivel	Base de Datos					
1.2.	Nombre Unidad	Casos prácticos					
1.3.	Tema 1 – Unidad	Práctica de Comandos DML y DDL					
1.4.	Tema 2 – Unidad	Introducción a Seguridad en Base de Datos					
1.5.	Unidad organizacional	Segundo					
1.6.	Unidad	4					
1.7.	Total, Horas Unidad	15					
1.8.	Detalle de horas Unidad	ACD	6	AA	4	APE	5

Resultado de Aprendizaje

Al finalizar esta unidad, los estudiantes serán capaces de recuperar, insertar, actualizar y eliminar información de una base de datos utilizando Transact/SQL y aplicando múltiples condiciones de comparación o funciones para el manejo de campos tipo fecha. construir programas estructurados y manejar los errores utilizando el lenguaje Transact-SQL.

Metodología

La Metodología de se basa en un enfoque activo y colaborativo. Incluye sesiones de clase estructuradas en torno a temas específicos, lecturas y tareas, estudio de casos, proyectos prácticos y debates en clase. Los estudiantes también se involucran en trabajos en grupo y mantienen diarios de reflexión. Se fomenta la participación en debates y preguntas y respuestas, y se proporcionan recursos en línea para una comprensión más profunda. La evaluación se realiza mediante pruebas, proyectos y una evaluación final, con retroalimentación continua para guiar el aprendizaje. Esta metodología busca que los estudiantes comprendan los conceptos fundamentales y puedan aplicar eficazmente las metodologías de desarrollo de software en situaciones prácticas.

DESARROLLO

Unidad 4: Casos prácticos

Tema 1: Práctica de Comandos DML y DDL

Objetivo: Aplicar comandos DML (Data Manipulation Language) y DDL (Data Definition Language) en la gestión de bases de datos, comprendiendo su uso en la manipulación, modificación y estructuración de la información, con el propósito de desarrollar sistemas de bases de datos eficientes y bien estructurados.

Desarrollo de casos prácticos. DML y DDL

El desarrollo de casos prácticos que involucra comandos Data Definition Language (DDL) y Data Manipulation Language (DML) en bases de datos es esencial para consolidar el conocimiento y las habilidades de los estudiantes. Aquí, proporcionaré información detallada, ejemplos, comparaciones y referencias para abordar este tema:

Definición:

Casos Prácticos: Se refieren a situaciones simuladas o problemas del mundo real que los estudiantes deben resolver aplicando conocimientos específicos, en este caso, relacionados con la manipulación y definición de datos en bases de datos.

Ejemplo de Caso Práctico:

Supongamos que, en el contexto de una biblioteca, se requiere desarrollar un caso práctico para gestionar el inventario de libros. El caso podría incluir la creación de tablas (DDL) para almacenar información sobre libros, autores y categorías, así como la manipulación de datos (DML) para agregar nuevos libros, actualizar información existente, y realizar consultas para obtener información específica.

Comparación de DDL y DML:

DDL (Data Definition Language): Se utiliza para definir la estructura de la base de datos, incluyendo la creación de tablas, la especificación de claves primarias y foráneas, la definición de índices, entre otros.

Ejemplo DDL:

```
CREATE TABLE Libros (
  ISBN VARCHAR(13) PRIMARY KEY,
  Titulo VARCHAR(255),
  AutorID INT,
  CategoriaID INT,
  FOREIGN KEY (AutorID) REFERENCES Autores(AutorID),
  FOREIGN KEY (CategoriaID) REFERENCES Categorias(CategoriaID)
);
```

DML (Data Manipulation Language): Se utiliza para manipular los datos almacenados en la base de datos, como la inserción, actualización y eliminación de registros, así como la realización de consultas para recuperar información.

Ejemplo DML:

```
-- Inserción de un nuevo libro
INSERT INTO Libros (ISBN, Titulo, AutorID, CategoriaID)
VALUES ('1234567890123', 'Ejemplo de Libro', 1, 2);

-- Actualización del título de un libro
UPDATE Libros
SET Titulo = 'Nuevo Título'
WHERE ISBN = '1234567890123';

-- Eliminación de un libro
DELETE FROM Libros
WHERE ISBN = '1234567890123';
```

-- Consulta para obtener información

SELECT * FROM Libros WHERE CategoriaID = 2;

Metodología para Desarrollar Casos Prácticos:

Definición de Requerimientos: Identificar los requisitos específicos del sistema de gestión de la unidad de Biblioteca, incluyendo la información que se debe almacenar, relaciones entre entidades, y operaciones a realizar.

Diseño de la Estructura de la Base de Datos (DDL): Crear un modelo de datos que refleje la estructura de la base de datos, especificando tablas, claves primarias y foráneas, así como otros elementos necesarios.

Desarrollo de Casos Prácticos (DML): Diseñar situaciones prácticas que involucren la manipulación de datos, como la inserción, actualización, eliminación y consulta de información. Estos casos deben alinearse con los requisitos establecidos.

Implementación en un Entorno de Desarrollo: Utilizar un entorno de desarrollo SQL para ejecutar y probar los comandos DDL y DML, asegurándose de que el sistema cumple con los requisitos.

Optimización y Refinamiento: Revisar y optimizar los comandos SQL para garantizar un rendimiento eficiente del sistema, considerando índices y otras estrategias de optimización.

Evaluación y Retroalimentación: Evaluar la ejecución de los casos prácticos, proporcionar retroalimentación a los estudiantes y permitir oportunidades de mejora.

Al seguir esta metodología, los estudiantes no solo aprenderán los comandos DDL y DML, sino que también desarrollarán habilidades prácticas aplicables a situaciones del mundo real en el contexto de la gestión de una biblioteca.

Transacciones y Técnicas de Recuperación

Definición:

Transacciones: En el contexto de bases de datos, una transacción es una secuencia de una o más operaciones de base de datos que deben ejecutarse de manera indivisible. Una transacción garantiza la integridad y consistencia de la base de datos, y sigue el principio ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

Principio ACID:

Atomicidad: Una transacción se considera atómica si todas sus operaciones se ejecutan de manera completa o ninguna se ejecuta en absoluto. No hay estado intermedio.

Consistencia: Después de que una transacción se ha completado, la base de datos debe estar en un estado consistente. Cualquier cambio debe cumplir con las reglas de integridad definidas.

Aislamiento: Cada transacción debe ser independiente de otras transacciones en ejecución. Los resultados de una transacción no deben ser visibles para otras transacciones hasta que se complete.

Durabilidad: Los cambios realizados por una transacción exitosa deben ser permanentes incluso en caso de fallo del sistema. La base de datos debe ser duradera.

Ejemplo de Transacción:

```
BEGIN TRANSACTION;

UPDATE Cuentas SET Saldo = Saldo - 100 WHERE UsuarioID = 1;

INSERT INTO RegistroTransacciones (UsuarioID, Monto) VALUES (1, -100);

COMMIT;
```

En este ejemplo, la transacción reduce el saldo de un usuario y registra la transacción en una tabla de historial. Si alguna operación falla, la transacción se revertirá automáticamente (ROLLBACK).

Técnicas de Recuperación:

Puntos de Control (Checkpoints): Consiste en tomar instantáneas periódicas del estado de la base de datos. Si ocurre un fallo, la recuperación puede comenzar desde el último punto de control.

Registro de Transacciones (Transaction Logging): Todas las operaciones de transacción se registran en un archivo de registro antes de aplicarse a la base de datos. En caso de fallo, las transacciones incompletas se pueden deshacer o completar desde el archivo de registro.

Comparación:

Puntos de Control vs. Registro de Transacciones:

Los puntos de control son instantáneas periódicas que ayudan en la recuperación, pero si se pierde información después de un punto de control, la recuperación puede ser más difícil.

El registro de transacciones garantiza que todas las operaciones estén registradas antes de aplicarse, facilitando la recuperación hasta el último estado conocido.

Importancia:

Las transacciones y las técnicas de recuperación son fundamentales para garantizar la integridad y la disponibilidad de la información en bases de datos. Su comprensión y aplicación adecuadas son esenciales para desarrolladores y administradores de bases de datos.

Procesamiento de transacciones, COMMIT, ROLLBACK.

Definición:

Procesamiento de Transacciones: Es el conjunto de operaciones realizadas en una base de datos como parte de una transacción. Involucra la ejecución de una o más operaciones SQL que deben ser tratadas como una unidad atómica.

Commit y Rollback:

COMMIT: Es un comando utilizado para confirmar que todas las operaciones realizadas dentro de una transacción se completen con éxito y que los cambios deben hacerse permanentes en la base de datos.

Ejemplo de COMMIT:

```
BEGIN TRANSACTION;

UPDATE Cuentas SET Saldo = Saldo - 100 WHERE UsuarioID = 1;

INSERT INTO RegistroTransacciones (UsuarioID, Monto) VALUES (1, -100);

COMMIT;
```

En este ejemplo, si todas las operaciones dentro de la transacción son exitosas, el COMMIT confirma los cambios en la base de datos.

ROLLBACK: Es un comando utilizado para deshacer todas las operaciones realizadas dentro de una transacción que aún no han sido confirmadas. Se utiliza en situaciones de error o cuando se desea revertir una transacción.

Ejemplo de ROLLBACK:

```
BEGIN TRANSACTION;

UPDATE Cuentas SET Saldo = Saldo - 100 WHERE UsuarioID = 1;

INSERT INTO RegistroTransacciones (UsuarioID, Monto) VALUES (1, -100);

-- En caso de error, deshacer la transacción

ROLLBACK;
```

Si ocurre algún error, el ROLLBACK deshace todas las operaciones realizadas dentro de la transacción.

Comparación:

COMMIT vs. ROLLBACK:

COMMIT: Confirma y hace permanentes los cambios realizados durante una transacción exitosa.

ROLLBACK: Deshace y revierte todas las operaciones de una transacción, asegurando que no se realicen cambios no deseados en la base de datos en caso de error.

Importancia:

El procesamiento de transacciones, junto con los comandos COMMIT y ROLLBACK, es crucial para garantizar la integridad y consistencia de los datos en una base de datos. La correcta gestión de transacciones asegura que los cambios se realicen de manera controlada y reversible en caso de errores.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Consideras importante la tarea de recuperación en una base de datos. Detalla y justifica tu respuesta.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Configurar conexiones a motor de base de datos.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.
2. Observa el video de la clase magistral.
3. Revisa en el calendario el día y hora en que se debe de realizar el Test.
4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

Tema 2: Introducción a Seguridad en Base de Datos

Definición:

Seguridad en Bases de Datos: Se refiere a la implementación de medidas y controles para proteger la confidencialidad, integridad y disponibilidad de los datos almacenados en una base de datos. Incluye aspectos como la autenticación, autorización, cifrado y auditoría.

Componentes Clave de la Seguridad en Bases de Datos

Autenticación

Definición: Es el proceso de verificar la identidad de un usuario o sistema que intenta acceder a la base de datos.

Ejemplo: Utilización de nombres de usuario y contraseñas.

Comparación: La autenticación de dos factores proporciona una capa adicional de seguridad al requerir más de una forma de identificación.

Autorización

Definición: Implica otorgar o negar permisos a usuarios específicos para realizar ciertas operaciones en la base de datos.

Ejemplo: Conceder permisos de lectura, escritura o eliminación a roles o usuarios específicos.

Comparación: La autorización diferencial permite definir permisos basados en roles, simplificando la administración de permisos.

Cifrado de Datos

Definición: Consiste en convertir datos en un formato ilegible para aquellos que no tienen la clave de cifrado.

Ejemplo: Cifrado de contraseñas almacenadas en la base de datos.

Comparación: El cifrado a nivel de columna permite cifrar datos específicos en lugar de la base de datos completa.

Auditoría

Definición: Registro y seguimiento de las actividades de la base de datos para detectar comportamientos inusuales o posibles amenazas.

Ejemplo: Registro de eventos como intentos de acceso no autorizados.

Comparación: La auditoría continua proporciona un monitoreo constante en tiempo real.

Importancia de la Seguridad en Bases de Datos

Confidencialidad: Garantiza que solo usuarios autorizados puedan acceder a la información confidencial.

Integridad: Asegura que los datos sean precisos y no se vean comprometidos por cambios no autorizados.

Disponibilidad: Garantiza que la base de datos esté disponible y funcional cuando sea necesario.

Ejemplo Práctico:

En el contexto de bases de datos, los "objetos" son elementos fundamentales

```
-- Creación de un usuario con permisos limitados
CREATE USER 'usuario_seguro'@'localhost' IDENTIFIED BY 'contrasena_segura';
GRANT SELECT ON base_de_datos.tabla1 TO 'usuario_seguro'@'localhost';
```

En este ejemplo, se crea un usuario con una contraseña segura y se le concede permiso solo para realizar operaciones SELECT en la tabla1 de la base_de_datos.

La seguridad en bases de datos es esencial para proteger la información sensible y mantener la integridad de los datos almacenados. Al implementar medidas de seguridad adecuadas, se contribuye a prevenir accesos no autorizados y posibles amenazas.

Implementación de Scripts, aplicación mediante SQL.

La implementación de scripts mediante SQL para la seguridad en bases de datos implica la creación, gestión y asignación de permisos a usuarios y roles utilizando instrucciones SQL. Esto garantiza un control preciso sobre quién puede acceder a qué datos y realizar qué operaciones en la base de datos.

Componentes Clave de la Implementación de Scripts

Creación de Usuarios

Definición: La creación de usuarios establece identidades que pueden acceder a la base de datos.

Ejemplo: CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'contrasena';

Concesión de Privilegios

Definición: Consiste en otorgar permisos específicos a usuarios o roles para realizar operaciones en tablas u objetos de la base de datos.

Ejemplo: GRANT SELECT, INSERT ON tabla1 TO 'nombre_usuario'@'localhost';

Creación de Roles

Definición: Los roles son conjuntos de privilegios que se pueden asignar a usuarios, simplificando la gestión de permisos.

Ejemplo: CREATE ROLE 'rol_lectura';

Asignación de Roles a Usuarios

Definición: Consiste en asignar roles a usuarios para agrupar permisos de manera eficiente.

Ejemplo: GRANT 'rol_lectura' TO 'nombre_usuario'@'localhost';

Revocación de Privilegios

Definición: Permite retirar permisos previamente concedidos.

Ejemplo: REVOKE INSERT ON tabla1 FROM 'nombre_usuario'@'localhost';

Comparación

Implementación de Scripts vs. Herramientas Visuales

Implementación de Scripts: Proporciona un control preciso y detallado sobre los permisos, ideal para escenarios complejos y personalizados.

Herramientas Visuales: Ofrecen interfaces gráficas que facilitan la gestión, pero pueden ser menos flexibles en comparación con scripts personalizados.

Ejemplo Práctico:

```
-- Creación de un usuario
CREATE USER 'empleado'@'localhost' IDENTIFIED BY 'contrasena_segura';

-- Concesión de privilegios SELECT e INSERT en una tabla
GRANT SELECT, INSERT ON base_de_datos.tabla1 TO 'empleado'@'localhost';

-- Creación de un rol
CREATE ROLE 'rol_empleado';

-- Asignación de rol a usuario
GRANT 'rol_empleado' TO 'empleado'@'localhost';
```

En este ejemplo, se crea un usuario 'empleado', se le conceden privilegios en la tabla1 y se crea un rol 'rol_empleado' que se asigna al usuario 'empleado'. Esto facilita la gestión de permisos en grupos específicos.

Accesos autorizados: Roles y Privilegios

Roles en Base de Datos: Los roles son conjuntos de privilegios que se pueden asignar a usuarios. Permiten agrupar permisos de manera eficiente y simplificar la administración de accesos autorizados.

Privilegios en Base de Datos: Son permisos específicos que se conceden a usuarios o roles para realizar operaciones sobre objetos de la base de datos. Estos pueden incluir SELECT, INSERT, UPDATE, DELETE, entre otros.

Roles

Los roles son una herramienta eficaz para la gestión de accesos autorizados en bases de datos. Algunos aspectos clave incluyen:

Creación de Roles

Ejemplo: CREATE ROLE 'rol_lectura';

Definición: Los roles se crean para representar conjuntos de privilegios específicos.

Asignación de Privilegios a Roles

Ejemplo: GRANT SELECT ON tabla1 TO 'rol_lectura';

Definición: Los privilegios se asignan a roles, permitiendo que los usuarios que tienen ese rol obtengan automáticamente esos privilegios.

Asignación de Roles a Usuarios

Ejemplo: GRANT 'rol_lectura' TO 'usuario_1'@'localhost';

Definición: Los roles se asignan a usuarios individuales, simplificando la administración.

Privilegios

La concesión de privilegios es esencial para definir qué operaciones pueden realizar los usuarios en la base de datos:

Concesión de Privilegios a Usuarios Directamente

Ejemplo: GRANT INSERT ON tabla1 TO 'usuario_2'@'localhost';

Definición: Los privilegios pueden concederse directamente a usuarios.

Revocación de Privilegios

Ejemplo: REVOKE DELETE ON tabla2 FROM 'usuario_3'@'localhost';

Definición: Permite retirar privilegios previamente concedidos.

Comparación

Roles vs. Privilegios Directos

Roles: Facilitan la asignación y gestión de conjuntos de privilegios. Útiles para asignar los mismos permisos a múltiples usuarios.

Privilegios Directos: Proporcionan una flexibilidad más granular al asignar permisos específicos a usuarios individuales.

Ejemplo Práctico:

```
-- Creación de un rol
CREATE ROLE 'rol_administrador';

-- Asignación de privilegios a un rol
GRANT SELECT, INSERT, UPDATE, DELETE ON base_de_datos.* TO 'rol_administrador';

-- Creación de un usuario y asignación de un rol
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'contrasena_segura';
GRANT 'rol_administrador' TO 'admin'@'localhost';
```

En este ejemplo, se crea un rol 'rol_administrador' al que se le otorgan privilegios completos sobre todas las tablas de la base de datos. Luego, se crea un usuario 'admin' al que se le asigna el rol 'rol_administrador', otorgándole automáticamente los privilegios asociados.

Implementación de Scripts de seguridad y autorización

La implementación de scripts de seguridad y autorización en bases de datos implica la creación y ejecución de scripts SQL para establecer medidas de seguridad, como la creación de usuarios, asignación de roles y concesión de privilegios. Esto garantiza un control preciso sobre quién puede acceder a la base de datos y qué operaciones pueden realizar.

Componentes Clave de la Implementación de Scripts

Creación de Usuarios

Ejemplo: CREATE USER 'usuario_seguro'@'localhost' IDENTIFIED BY 'contrasena_segura';

Definición: La creación de usuarios establece identidades que pueden acceder a la base de datos.

Asignación de Roles a Usuarios

Ejemplo: GRANT 'rol_lectura' TO 'usuario_seguro'@'localhost';

Definición: Los roles son conjuntos de privilegios que se asignan a usuarios para agrupar permisos de manera eficiente.

Concesión de Privilegios

Ejemplo: GRANT SELECT ON tabla1 TO 'usuario_seguro'@'localhost';

Definición: Los privilegios se otorgan a usuarios o roles para permitir operaciones específicas en tablas u objetos de la base de datos.

Revocación de Privilegios

Ejemplo: REVOKE INSERT ON tabla2 FROM 'usuario_seguro'@'localhost';

Definición: Permite retirar privilegios previamente concedidos a usuarios.

Comparación

Implementación de Scripts vs. Herramientas Visuales

Implementación de Scripts: Proporciona un control preciso y detallado sobre los permisos. Ideal para escenarios complejos y personalizados.

Herramientas Visuales: Ofrecen interfaces gráficas que facilitan la gestión, pero pueden ser menos flexibles en comparación con scripts personalizados.

Ejemplo Práctico:

```
-- Creación de un usuario
CREATE USER 'empleado'@'localhost' IDENTIFIED BY 'contrasena_segura';

-- Concesión de privilegios SELECT e INSERT en una tabla
GRANT SELECT, INSERT ON base_de_datos.tabla1 TO 'empleado'@'localhost';

-- Creación de un rol
CREATE ROLE 'rol_empleado';

-- Asignación de rol a usuario
GRANT 'rol_empleado' TO 'empleado'@'localhost';
```

En este ejemplo, se crea un usuario 'empleado', se le conceden privilegios en la tabla1 y se crea un rol 'rol_empleado' que se asigna al usuario 'empleado'. Esto facilita la gestión de permisos en grupos específicos mediante scripts SQL.

Optimización de consultas

La optimización de consultas en bases de datos es el proceso de mejorar el rendimiento y la eficiencia de las consultas SQL. El objetivo es obtener resultados más rápidos y utilizar de manera eficiente los recursos del sistema.

Componentes Clave de la Optimización de Consultas

Uso de Índices

Ejemplo: CREATE INDEX idx_nombre ON tabla1(columna_nombre);

Definición: Los índices aceleran la recuperación de datos al proporcionar un acceso más rápido a las filas de una tabla.

Selección de Columnas Necesarias

Ejemplo: SELECT columna1, columna2 FROM tabla1 WHERE condicion;

Definición: Seleccionar solo las columnas necesarias reduce la cantidad de datos recuperados, mejorando la eficiencia.

Uso de Cláusulas WHERE y JOIN Eficientes

Ejemplo: SELECT * FROM tabla1 WHERE columna1 = 'valor' AND columna2 > 10;

Definición: Utilizar cláusulas WHERE bien construidas y unir tablas de manera eficiente mejora la precisión y rapidez de las consultas.

Evitar Funciones Costosas

Ejemplo: SELECT * FROM tabla1 WHERE YEAR(fecha_columna) = 2022;

Definición: Las funciones en columnas pueden hacer que las consultas sean más lentas. Es preferible realizar operaciones en valores constantes.

Comparación

Optimización Manual vs. Herramientas Automáticas

Optimización Manual: Permite un control preciso y adaptado a situaciones específicas, pero puede requerir un conocimiento profundo.

Herramientas Automáticas: Ofrecen soluciones rápidas y generalizadas, pero pueden no abordar problemas específicos de la aplicación.

Ejemplo Práctico:

```sql
-- Creación de un índice
CREATE INDEX idx_nombre ON empleados(nombre);

-- Consulta optimizada seleccionando columnas necesarias
SELECT id, nombre, salario FROM empleados WHERE departamento = 'Ventas';

-- Uso de cláusulas WHERE y JOIN eficientes
SELECT clientes.nombre, pedidos.fecha
FROM clientes
JOIN pedidos ON clientes.id = pedidos.cliente_id
WHERE pedidos.estado = 'Entregado';
```

En este ejemplo, se crea un índice en la columna 'nombre', se seleccionan solo las columnas necesarias en una consulta y se utiliza una cláusula JOIN eficiente para relacionar las tablas. Estas prácticas contribuyen a la optimización de consultas en la base de datos.

ACTIVIDADES PRÁCTICAS

Tabla

Detalle de la Práctica

Practicas	Instrucciones	Duración de la Practica	Mecanismo de Evaluación
Foro de debate	Revisar Actividades Prácticas	0:30 mm	Por rúbricas
Práctica investiga: Taller/Ensayo/Investigación	Revisar Actividades Prácticas	3:30 h	Por rúbricas
Tutorías Prácticas	Revisar Actividades Prácticas	2:00 mm	Por participación en clase
Test evaluativo	Revisar Actividades Prácticas	0:30 mm	Formativa

Nota. Elaboración propia

Consideraciones Generales

1. Lee y revisa la Guía de Estudio.
2. Observa el video de la clase magistral.
3. Participa activamente en el FORO DEBATE.
4. Revisa en el calendario el día y hora en que se realizará las actividades prácticas y la tutoría.
5. Realizar la práctica propuesta por el docente.
6. Revisa las Rúbricas acorde a la actividad prácticas.
7. Toma los principales apuntes que vayas rescatando del texto.
8. Reflexiona sobre los temas tratados en la clase.
9. Sube el documento en formato PDF en la plataforma Moodle.
10. Revisa la calificación de tus actividades prácticas.

FORO DE DEBATE

Tema: Importancia de la seguridad en una Base de Datos.

Instrucciones para Participar:

- Exponga su punto de vista acerca del tema.
- Comenta las respuestas de otros participantes.
- Promueve un ambiente respetuoso y constructivo.

PRÁCTICA INVESTIGATIVA

Taller: Configurar perfiles de usuarios de nivel 1 y nivel 2 para acceso a base de datos, según caso de estudio.

Consigna:

- El trabajo de investigación debe presentarse de manera estructurada y coherente para comunicar efectivamente los hallazgos y conclusiones.
- Asegúrese de que el estilo y formato del trabajo sigan las pautas de citación y presentación establecidas por la institución.
- Utilice un lenguaje claro y evite jergas o términos técnicos innecesarios.
- Revise y corrija la redacción y la gramática antes de la presentación final (ver anexo).

PRÁCTICA TEST

Antes de realizar el test, es importante considerar varios aspectos para aprovechar al máximo la experiencia.

Consideraciones para realizar el test

1. Lee y revisa el documento.
2. Observa el video de la clase magistral.
3. Revisa en el calendario el día y hora en que se debe de realizar el Test.
4. Realiza el Test de la unidad considerando lo Siguiente:

- No cambiarte entre paginas durante la realización del Test.
- Revisa que tu conexión a internet sea estable.
- Recuerda que todo el contenido está inmerso dentro del documento base.

5. Revisa tu puntuación, de inmediato al finalizar de la actividad.

V. Bibliografía

Bibliografía básica

Cabellero Muñoz - Reja, I. E. (2019). Calidad de Datos, 1 edición. RA-MA.

Teodoro, C. N., & Sara, A. T. (2023). Python desde el Laboratorio: Estructura de Datos, 1 edición. MACRO.

Bibliografía de consulta

Marqués, M. (2011). Bases de Datos. MetaBiblioteca. Obtenido de http://libros.metabiblioteca.org/handle/001/353

Ricardo, C. M. (2009). Bases de datos. McGranHill. Obtenido de http://up-rid2.up.ac.pa:8080/xmlui/handle/123456789/1354

Goméz, J. M. (2014). Diseño de bases de datos relacionados. España: Paraninfo. Obtenido de https://books.google.com.ec/books?id=EwcuBwAAQBAJ&printsec=frontcover&dq=base+de+datos&hl=es&sa=X&redir_esc=y#v=onepage&q=base%20de%20datos&f=false

Anexo 1: Modelo de Foro

Título del Foro:

[Nombre del Tema o Pregunta]

Desarrollo:

[Breve descripción del tema o pregunta]

Reglas del Foro:

Regla 1: Antes de participar en el debate, realiza una investigación exhaustiva sobre el tema en cuestión. Comprender los hechos, datos y argumentos relacionados con el tema fortalecerá tu posición y mejorarán la calidad de la discusión.

Regla 2: Muestra respeto hacia tus compañeros de debate, incluso si no estás de acuerdo con sus opiniones. Escucha activamente sus argumentos y evita interrupciones. Expresa tus puntos de vista de manera respetuosa y constructiva.

Regla 3: Contribuye de manera equitativa al debate, evitando la monopolización de la conversación. Respeta el tiempo asignado y asegúrate de que otros también tengan la oportunidad de expresar sus opiniones.

Anexo 2: Esquema de Ensayo

Caratula: (Escribe aquí los datos referentes a la materia, nombres completos, tema, periodo, Institución Educativa y logos)

Título:

(Escribe aquí el título de tu ensayo)

Introducción:

Desarrollo:

Conclusión:

Bibliografía

Anexo 3: Esquema de Taller/Investigación

Caratula:

(Escribe aquí los datos referentes a la materia, tema, periodo, Institución Educativa y logos)

Título:

(Escribe aquí el título principal de tu investigación)

Temas a investigar:

- **Tema 1:**
- **Tema 2:**
- **Tema 3:**

Conclusión:

Bibliografía

Anexos

Anexo 4: Rubrica de taller práctico

INDICADORES	MS 10 PUNTOS	S 6 PUNTOS	PS 2 PUNTOS	PUNTAJE
Número de ejercicios resueltos	Realiza en su totalidad los ejercicios que se propones.	Realiza entre la mitad y más de los ejercicios propuestos.	Realiza la menor parte o no realiza ningún ejercicio propuesto.	**40%**
Procedimiento y resultados de los ejercicios resueltos	Desarrolla el procedimiento, lo detalla y organiza los datos obtenidos.	Desarrolla el procedimiento, lo detalla poco y organiza algunos, pero otros no.	Detalla poco o no detalla nada del procedimiento.	**40%**
Presentación	Maneja una presentación prolija de la tarea.	La tarea se presenta de una forma aceptable.	La tarea no cuenta con una buena presentación	**20%**
			TOTAL=	100 %

Puntaje: 10 puntos

Anexo 5: Rúbrica de Foro

INDICADORES	MS 5 PUNTOS	S 3 PUNTOS	PS 2 PUNTOS	PUNTAJE
Comprensión del tema	La respuesta argumentada demuestra conocimiento concreto del tema tratado.	La respuesta argumentada demuestra conocimiento general de la mayoría del tema.	La respuesta argumentada demuestra un conocimiento limitado del tema.	20 %
Cuestionamientos y criterios	Presentó las dudas e inquietudes que se le presentaron durante el desarrollo del tema, o al menos sacó diferentes conclusiones, que las expuso en el debate.	Presenta al menos 2 preguntas o argumentos, aunque demuestra que no ha comprende ni revisado en profundidad el material.	No presenta dudas ni argumentos, aunque conozca del tema, no aporta al debate.	20 %
Redacción y ortografía	No presenta errores ortográficos ni de redacción.	Presenta algunos errores ortográficos y/o de redacción.	Presenta muchos errores de redacción y de ortografía.	20 %
Interacción: relación/ confrontación con intervenciones de otros/as	Comparte activamente con de sus compañeros los temas planteados y resuelve dudas de sus compañeros con argumentos valederos en más de 2 intervenciones.	Comparte con de sus compañeros los temas planteados, argumentando en al menos de 2 intervenciones.	Plantea dudas a sus compañeros con el tema, sin embargo, no participa ni responde con argumentos lógicos. Solo interviene una vez.	20 %
Participación	Participa con más de 2 intervenciones, además de su contestación.	Participa al menos con 5 Intervenciones, además de su contestación.	No interviene en otra participación a más de la suya.	20 %
			TOTAL=	100 %

Puntaje: 5 puntos

Anexo 6: Rubrica de Ensayo

INDICACIONES	MS 10 PUNTOS	S 6 PUNTOS	PS 2 PUNTOS	PUNTAJE
Enfoque o Idea principal	La idea principal nombra el tema del ensayo y esquematiza los puntos principales a discutir.	La idea principal nombra el tema del ensayo.	La idea principal esquematiza algunos o todos los puntos a discutir, pero no menciona el tema.	20 %
Transiciones o Conectores	Una variedad bien pensada de transiciones fue usada. Estas muestran claramente como están las ideas conectadas.	Las transiciones muestran cÃ³mo están las ideas conectadas, pero hay muy poca variedad.	Algunas transiciones funcionan bien, pero la conexiÃ³n entre las ideas no es clara.	20 %
Conclusión	La conclusión es fuerte y dejó al lector con una idea absolutamente clara de la posición del autor. Un parafraseo efectivo de la idea principal empieza la conclusión.	La conclusión es evidente. La posición del autor es parafraseada en las primeras dos oraciones de la conclusión.	La posición del autor es parafraseada en la conclusión, pero no al principio de la misma.	20 %
Gramática y ortografía	El autor no comete errores de gramática ni de ortografía que distraen al lector del contenido del ensayo.	El autor comete 1 a 2 errores gramaticales u ortográficos que distraen al lector del contenido del ensayo.	El autor comete 3 a 4 errores gramaticales u ortográficos que distraen al lector del contenido del ensayo.	20 %
Fuentes	Todas las fuentes usadas para las citas, las estadísticas y los hechos son creíbles y están citadas correctamente.	Todas las fuentes usadas para las citas, las estadísticas y los hechos son creíbles y la mayoría está citada correctamente.	La mayoría de las fuentes usadas para las citas, las estadísticas y los hechos es creíble y está citada correctamente.	20 %
			TOTAL=	100 %

Puntaje: 10 puntos

Anexo 7: Rúbrica de Taller/Investigación

INDICADORES	MS 10 PUNTOS	S 6 PUNTOS	PS 2 PUNTOS	PUNTAJE
Organización de las ideas	El texto es coherente y tiene sentido toda la lectura.	Presenta ideas aisladas, no enlazadas entre sí.	El texto no tiene coherencia y las ideas no están claras.	30 %
Ortografía	Tiene 4 o menos faltas ortográficas	Tiene entre 5 y 9 faltas ortográficas.	Tiene más de 10 faltas ortográficas.	30 %
Calidad de la información presentada	La información está claramente relacionada con el tema y proporciona varios niveles de relación con sus diferentes entornos.	La información da respuesta a la consigna principal del artículo, mas no aporta niveles de relaciones de la comunicación con el entorno próximo.	La información proporcionada guarda relación con las categorías, aunque su calidad argumentativa es deficiente.	40 %
			TOTAL=	100 %

Puntaje: 10 puntos