



UACJ

Título: Tarea 2

Docente: Jorge Isaac Arellano Flores

Materia: Bases de datos 1

Alumno: Ivan Gerardo Lomeli Reyna

Matricula: 204168

Fecha: 09/17/2023

1.- Defina los siguientes conceptos: información, campo, registro, archivo.

La información se refiere a los datos organizados y estructurados que se almacenan para su posterior acceso, consulta y análisis. Esta información se presenta en forma de registros y campos dentro de tablas, donde cada registro representa una entidad única y cada campo almacena un atributo específico de esa entidad. La información en una base de datos permite la gestión eficiente de datos, la toma de decisiones informadas y el soporte para diversas aplicaciones y procesos.

Un campo es una unidad básica de almacenamiento de datos. Representa una categoría específica de información dentro de un registro en una tabla. Cada campo almacena un tipo particular de dato, como texto, números, fechas, etc. Los campos se utilizan para organizar y estructurar los datos de manera que cada atributo de una entidad pueda ser capturado y almacenado de manera eficiente. Por ejemplo, en una base de datos de clientes, los campos podrían incluir el nombre, apellido, dirección, número de teléfono, etc. Cada registro en la tabla tendría valores específicos para cada uno de estos campos, lo que permite representar de manera completa la información de cada cliente.

Un registro es una colección de información relacionada que se almacena como una fila en una tabla. Representa una entidad única, como un objeto, una persona, un evento o cualquier cosa que se esté registrando en la base de datos. Cada registro consta de campos individuales que representan atributos específicos de esa entidad. Por ejemplo, en una base de datos de empleados, un registro podría contener campos como el nombre del empleado, su salario, su número de identificación y su fecha de contratación. En resumen, un registro es una entrada completa y coherente que almacena información específica sobre una entidad en una base de datos.

El término "**archivo**" no se utiliza de la misma manera que en sistemas de archivos tradicionales. En lugar de eso, se emplea el concepto de "tabla" para almacenar y organizar los datos de manera estructurada. Una tabla en una base de datos es un conjunto de registros, cada uno con sus campos correspondientes. Sin embargo, en sistemas de archivos tradicionales, un archivo es una unidad de almacenamiento que contiene datos relacionados, como documentos de texto, imágenes, videos, etc. Estos archivos se organizan en carpetas o directorios para facilitar la gestión y el acceso. Es importante destacar que las bases de datos y los sistemas de archivos son dos enfoques diferentes para almacenar y gestionar información, y sus terminologías y estructuras varían en función de sus propios paradigmas.

2.- ¿Qué es un Cliente, y que un Servidor?

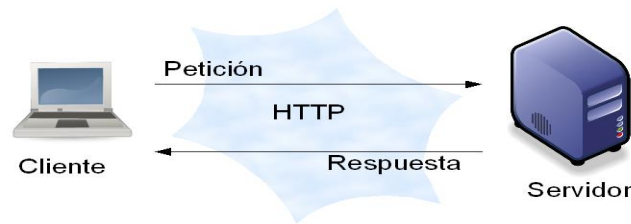
El modelo cliente-servidor tiene algunos rasgos característicos. Hay una clara distribución de tareas entre los clientes y los servidores. El servidor es el responsable de proporcionar los servicios. Se encarga de ejecutar los servicios solicitados y entrega la respuesta esperada. El cliente, en cambio, utiliza y solicita los servicios proporcionados. Finalmente, recibe la respuesta del servidor.

En el modelo cliente-servidor, un servidor sirve a varios clientes y, por ende, procesa múltiples peticiones de diferentes clientes. Para ello, presta su servicio de forma permanente y pasiva. Por su parte, el cliente solicita activamente los servicios del servidor e inicia las tareas del servidor.

Un servidor es un sistema que contiene datos o proporciona recursos a los que deben acceder otros sistemas de la red. Un cliente es un sistema que solicita servicios o datos de un servidor.

Los tipos de servidor comunes son servidores de archivos que almacenan archivos, servidores de nombres que almacenan nombres y direcciones, servidores de aplicaciones que almacenan programas y aplicaciones y servidores de impresión que planifican y dirigen los trabajos de impresión al destino.

Un cliente puede solicitar código de programa actualizado o el uso de aplicaciones de un servidor de código. Para obtener un nombre o una dirección, un cliente se pone en contacto con un servidor de nombres. Un cliente también puede solicitar archivos y datos para la entrada de datos, las consultas o la actualización de registros de un servidor de archivos.



En la imagen se representa el modelo cliente servidor, es decir un cliente realiza una petición a la espera de una respuesta del servidor, ya sea en forma de estatus o JSON.

3.- ¿Cuáles son los tipos de Clientes y de Servidores se tienen? De una breve descripción de estos.

Clientes:

Cliente ligero:

Es una computadora cliente en una arquitectura de red cliente-servidor que depende del servidor central para las tareas de procesamiento, y se dedica principalmente a transportar la entrada y la salida entre el usuario y el servidor remoto.

Cliente pesado:

Es una computadora cliente en una arquitectura de red cliente-servidor que realiza la mayor parte de la carga de cómputo, transmitiendo al servidor únicamente los datos.

Cliente híbrido:

Es una computadora cliente sin unidad de disco, por lo que emplea el arranque por red para cargar su sistema operativo desde un servidor. Los clientes híbridos procesan los datos utilizando su propia CPU y memoria RAM para ejecutar el software, pero no almacenan los datos persistentemente, esa tarea la realiza el servidor.

Servidores:

Servidor de impresiones: controla una o más impresoras y acepta trabajos de impresión de otros clientes de la red, poniendo en cola los trabajos de impresión.

Servidor de correo: almacena, envía, recibe, enruta y realiza otras operaciones relacionadas con email para los clientes de la red.

Servidor de fax: almacena, envía, recibe, enruta y realiza otras funciones necesarias para la transmisión, la recepción y la distribución apropiadas de los fax.

Servidor de la telefonía: realiza funciones relacionadas con la telefonía, como es la de contestador automático, almacenando los mensajes de voz, encaminando las llamadas y controlando también la red o el Internet.

Servidor proxy: realiza un cierto tipo de funciones a nombre de otros clientes en la red para aumentar el funcionamiento de ciertas operaciones (p. ej., prefetching y depositar documentos u otros datos que se soliciten muy frecuentemente), también proporciona servicios de seguridad, o sea, incluye un cortafuegos.

Servidor del acceso remoto (RAS): controla las líneas de módem de los monitores u otros canales de comunicación de la red para que las peticiones conecten con la red de una posición remota, responde llamadas telefónicas entrantes y reconoce la petición de la red.

Servidor web: almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material Web compuesto por datos (conocidos colectivamente como contenido), y distribuye este contenido a clientes que la piden en la red.

Servidor de base de datos: provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio.

Servidor de Seguridad: Tiene software especializado para detener intrusiones maliciosas, normalmente tienen antivirus, antispyware, antiadware, además de contar con cortafuegos redundantes de diversos niveles y/o capas para evitar ataques, los servidores de seguridad varían dependiendo de su utilización e importancia.

4.- ¿Qué es una Base de Datos?

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

5.- Características, objetivos y funciones principales de las bases de datos.

Características: Integridad, Consistencia, independencia de datos

Integridad:

El término integridad de datos se refiere a la corrección y exactitud de los datos en una base de datos.

Consistencia:

El concepto de consistencia nos lleva necesariamente al de integridad, ya que tiene que ver con el estado de la información: correcta o incorrecta.

En los casos en los que no se ha logrado eliminar la redundancia (almacenamiento de los mismos datos varias veces en diferentes lugares), será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Independencia de datos:

Se conoce como independencia de datos a la capacidad para utilizar la base de datos sin conocer los detalles de su representación conceptual (Independencia lógica de datos) o de su almacenamiento físico en el hardware (Independencia física de datos), es decir, la independencia de los datos consiste en la capacidad de modificar la presentación (física o lógica) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Objetivos:

Almacenar gran número de información de una forma organizada para su futura consulta, realización de búsquedas, nuevo ingreso de datos, etc. Todo esto lo permite realizar de una forma rápida y simple desde un ordenador. Fila: cada una conforma un registro.

Funciones principales:

Almacenamiento de Datos: Proporcionar un lugar estructurado para guardar información de manera eficiente.

Recuperación de Datos: Facilitar la búsqueda y recuperación de información específica mediante consultas.

Mantenimiento de Integridad: Garantizar la precisión y consistencia de los datos a través de validaciones.

Seguridad de Datos: Establecer niveles de acceso y permisos para proteger la información.

Apoyo a Transacciones: Manejar operaciones como unidades indivisibles para mantener la consistencia.

Generación de Informes: Crear informes y análisis para la toma de decisiones.

Escalabilidad y Rendimiento: Manejar grandes volúmenes de datos y usuarios sin perder eficiencia.

6.- Defina que es DBMS, para que sirve y algunos ejemplos comerciales.

Un sistema gestor de base de datos (SGBD) o Database Management System (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos.

Los gestores de datos o gestores de base de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones.

Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes.

Además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión.

Ejemplos de DBMS:

- Oracle.
- MySQL.
- Microsoft SQL Server.
- PostgreSQL.
- MongoDB.
- Redis.
- FireBird
- Apache Cassandra

7.- Explique que es el DBA y que funciones tiene.

La función principal de un DBA es implementar, mantener, optimizar y administrar estructuras de bases de datos para la organización. Básicamente, el DBA mantiene las bases de datos y las aplicaciones funcionando hasta PAR, o rendimiento, disponibilidad y capacidad de recuperación.

El DBA es el punto de contacto central para obtener información sobre los sistemas de administración de bases de datos que utiliza la organización. Esto significa mantenerse actualizado sobre las últimas versiones y capacidades de cada DBMS que administran. También implica educar e informar a los desarrolladores de aplicaciones y otros usuarios sobre cómo utilizar el DBMS y sus instalaciones.

8.- Defina DDL y ejemplos de donde se utiliza.

DDL (Data Definition Language) es un subconjunto de SQL (Lenguaje de consulta estructurado) que se utiliza para definir la estructura de una base de datos y sus objetos, como tablas, vistas, índices y procedimientos. DDL se utilizan para crear, modificar y eliminar objetos de bases de datos, como tablas, vistas, índices y procedimientos almacenados. Algunas de las sentencias DDL más comunes son:

CREATE: Esta sentencia crea un nuevo objeto de base de datos, como una tabla, una vista o un índice. Por ejemplo, la siguiente sentencia SQL crea una tabla llamada "clientes":

CREATE TABLE clientes (id INT PRIMARY KEY, name VARCHAR (255), address VARCHAR (255));

ALTER: Esta sentencia se utiliza para modificar un objeto de base de datos existente. Por ejemplo, la siguiente sentencia SQL añade una nueva columna llamada "email" a la tabla "customers":

ALTER TABLE clientes ADD correo electrónico VARCHAR (255);

DROP: Esta sentencia se utiliza para eliminar un objeto de base de datos existente. Por ejemplo, la siguiente sentencia SQL elimina la tabla "clientes":

DROP TABLE clientes;

TRUNCATE: Esta sentencia se utiliza para eliminar todas las filas de una tabla, pero a diferencia de la sentencia DROP, conserva la estructura y los índices de la tabla.

RENAME: Esta sentencia se utiliza para cambiar el nombre de un objeto de base de datos existente. Por ejemplo, la siguiente sentencia SQL cambia el nombre de la tabla "clientes" a "clientes":

RENAME TABLE clientes TO clientes;

9.- Defina que es la Abstracción de la Información y los niveles que tiene.

La abstracción de datos es la característica de un sistema de bases de datos, que permite al usuario o programador operar con los datos sin necesidad de conocer detalles que para él no son de "importancia", ofreciendo así una visión abstracta de estos. Para cumplir con tal fin se han definido diferentes niveles de abstracción.

Nivel Físico. Determina como están almacenados físicamente los datos (pistas, sectores, cilindros), representa el nivel más bajo.

Nivel Lógico o Conceptual. Determina la organización de los archivos. Índices, llaves, orden de campos, relaciones, tipos de datos.

Nivel de Vistas. Oculta parte de la información a los usuarios, es decir hace visible solo una parte de la base de datos.

Modelos de Datos:

Los modelos de datos son una colección de herramientas conceptuales para describir los datos y las relaciones entre ellos.

Modelo de Red:

Colecciones de registros y las relaciones entre datos se representan mediante enlaces

Modelo Jerárquico:

Los registros se organizan como colecciones de árboles, en lugar de grafos dirigidos

Modelo Relacional:

Es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores

10.-Defina y describa todos los Modelos de Datos.

Los modelos de datos son estructuras conceptuales que ayudan a organizar y representar la información dentro de un sistema de información, base de datos o cualquier otro contexto en el que se manejen datos. Estos modelos proporcionan una forma de comprender cómo se almacenan, se relacionan y se accede a los datos. Aquí te presento algunos de los modelos de datos más comunes:

Modelo de Datos Jerárquico:

En este modelo, los datos se organizan en una estructura jerárquica similar a un árbol. Cada registro tiene un único registro padre y puede tener múltiples registros hijos. Fue uno de los primeros en utilizarse, pero su rigidez en cuanto a las relaciones entre datos lo hizo menos flexible que otros modelos.

Modelo de Datos en Red:

Este modelo es una evolución del jerárquico. Permite que un registro tenga varios padres y varios hijos, lo que lo hace más flexible para representar relaciones complejas. Sin embargo, su complejidad y dificultad para ser entendido llevaron a su reemplazo por modelos más modernos.

Modelo Relacional:

El modelo relacional es uno de los más ampliamente utilizados. Se basa en tablas con filas y columnas, donde cada tabla representa una entidad y las filas son las instancias de esa entidad. Las relaciones entre entidades se establecen mediante claves primarias y foráneas. Es conocido por su simplicidad y flexibilidad.

Modelo de Datos Entidad-Relación (ER):

Este modelo se basa en la representación gráfica de entidades (objetos, conceptos) y sus relaciones. Emplea diagramas que muestran entidades, atributos y las conexiones entre ellas. Es útil para diseñar bases de datos y entender las relaciones en un nivel alto.

Modelo de Datos de Objetos:

Este modelo combina características de la programación orientada a objetos con la gestión de bases de datos. Permite la representación directa de objetos y sus métodos dentro de la base de datos. Es útil para aplicaciones que necesitan manejar objetos complejos y sus relaciones.

Modelo de Datos Semánticos:

Este modelo se centra en la representación de significado y semántica en los datos. Utiliza ontologías para definir conceptos y relaciones entre ellos. Es utilizado en la web semántica y en aplicaciones donde es importante entender el contexto y la semántica de los datos.

Modelo Dimensional:

Principalmente utilizado en el diseño de almacenes de datos y sistemas de soporte de decisiones, el modelo dimensional organiza los datos en torno a hechos y dimensiones. Los hechos son los eventos medibles (por ejemplo, ventas) y las dimensiones son las categorías que describen esos hechos (por ejemplo, tiempo, ubicación).

Modelo NoSQL:

Los modelos NoSQL son una categoría diversa que incluye varios modelos de datos, como documentos, grafos, columnas y clave-valor. Son utilizados para casos en los que los modelos relacionales pueden no ser la mejor opción debido a la escalabilidad y la flexibilidad necesarias.

11.- Defina que es Instancia de la Base de Datos.

Una instancia del Motor de base de datos funciona como un servicio que controla todas las solicitudes de aplicación para trabajar con datos de cualquiera de las bases de datos administradas por dicha instancia. Es el destino de las solicitudes de conexión (inicios de sesión) de aplicaciones. La conexión se ejecuta en una conexión de red si la aplicación y la instancia están en equipos independientes. Si la aplicación y la instancia están en el mismo equipo, la conexión de SQL Server se puede ejecutar como una conexión de red o una conexión en memoria.

12.- Defina Esquemas.

Un esquema de base de datos define cómo se organizan los datos dentro de una base de datos relacional. Esto incluye restricciones lógicas como, por ejemplo, nombres de tablas, campos, tipos de datos y las relaciones entre estas entidades. Los esquemas suelen utilizar representaciones visuales para comunicar la arquitectura de la base de datos, convirtiéndose en la base de la disciplina de gestión de datos de una organización. Este proceso de diseño de esquemas de bases de datos también se conoce como modelado de datos.

13.- ¿Qué es Reorganizar Datos?

Reorganizar datos en bases de datos se refiere al proceso de reestructurar y optimizar la organización física de los datos almacenados en una base de datos con el objetivo de mejorar el rendimiento y la eficiencia en las consultas y operaciones. Esta optimización puede llevarse a cabo a nivel de almacenamiento en disco o memoria, y busca minimizar la fragmentación, maximizar el acceso rápido a los datos y reducir el tiempo de respuesta de las consultas.

Las bases de datos almacenan datos en estructuras como tablas, índices y vistas, y estas estructuras pueden experimentar problemas de rendimiento con el tiempo debido a la inserción, actualización y eliminación constante de datos. Algunas de las técnicas comunes para reorganizar datos incluyen:

Reorganización de índices: Los índices son estructuras que permiten un acceso más rápido a los datos en una tabla. Con el tiempo, los índices pueden volverse fragmentados y afectar negativamente el rendimiento de las consultas. La reorganización de índices implica reorganizar las páginas de datos y reordenar las entradas del índice para reducir la fragmentación y mejorar la eficiencia en la búsqueda.

Reorganización de tablas: Cuando las tablas experimentan muchas operaciones de inserción, actualización y eliminación, es posible que los datos se dispersen en el espacio físico de almacenamiento. Esto puede llevar a problemas de fragmentación, donde los datos relacionados se almacenan en ubicaciones dispersas, lo que afecta el rendimiento de las consultas. La reorganización de tablas implica reorganizar los datos para que estén más contiguos y se acceda más eficientemente.

Compactación: En algunos sistemas de bases de datos, los espacios vacíos que quedan después de eliminar datos pueden llevar a una asignación ineficiente de espacio en disco. La compactación implica la eliminación física de estos espacios vacíos y la reorganización de los datos restantes para liberar espacio.

Reindexación: Los índices pueden volverse ineficientes con el tiempo debido a la actualización constante de datos. La reindexación implica recrear los índices desde cero para optimizar su estructura y mejorar la eficiencia.

Estadísticas: Mantener estadísticas precisas sobre los datos en una base de datos es esencial para que el optimizador de consultas tome decisiones informadas sobre cómo ejecutar consultas de manera eficiente. La reorganización de estadísticas implica actualizar y refrescar estas estadísticas para mejorar la calidad de las decisiones del optimizador.

En resumen, reorganizar datos en bases de datos es un proceso importante para mantener un rendimiento óptimo a medida que los datos cambian con el tiempo. La frecuencia y los métodos específicos de reorganización dependen del sistema de gestión de bases de datos utilizado y de las características particulares de la base de datos en cuestión.

14.- ¿Qué es Reestructurar Datos?

Reestructurar datos significa reorganizar la disposición o formato de los datos para mejorar su organización, accesibilidad y eficiencia en operaciones y consultas.

15.- Defina que es el Diccionario de Datos.

Un diccionario de datos es un componente esencial en la gestión de bases de datos que proporciona una descripción detallada de los datos almacenados en una base de datos. Contiene metadatos que describen la estructura, definición, relaciones y propiedades de los elementos de datos, como tablas, columnas, índices, restricciones y otros objetos de la base de datos. El diccionario de datos sirve como referencia centralizada y documentación para los desarrolladores, administradores de bases de datos y usuarios finales, facilitando la comprensión y el uso correcto de los datos en la base de datos.

16.- Defina DML y ejemplos de donde se utiliza.

DML (Data Manipulation Language) es un subconjunto de SQL (lenguaje de consulta estructurado) que se utiliza para manipular los datos de una base de datos. DML Las sentencias SQL se utilizan para insertar, actualizar y eliminar datos en una base de datos. Algunas de las sentencias DML más comunes son:

SELECT: Esta sentencia se utiliza para recuperar datos de una o varias tablas de una base de datos. Por ejemplo, la siguiente consulta SQL recupera todos los registros de la tabla "clientes":

SELECT * FROM clientes;

INSERT: Esta sentencia se utiliza para insertar nuevos datos en una tabla. Por ejemplo, la siguiente sentencia SQL inserta una nueva fila en la tabla "clientes":

INSERT INTO customers (id, name, address) VALUES (1, 'John Smith,' '123 Main St');

UPDATE: Esta sentencia se utiliza para modificar los datos existentes en una tabla. Por ejemplo, la siguiente sentencia SQL actualiza la dirección del cliente con un ID de 1 en la tabla "clientes":

UPDATE clientes SET dirección = '456 Park Ave' WHERE id = 1;

DELETE: Esta sentencia se utiliza para eliminar datos de una tabla. Por ejemplo, la siguiente sentencia SQL elimina el cliente con un ID de 1 de la tabla "clientes":

DELETE FROM clientes WHERE id = 1;

DML Las sentencias SQL se ejecutan inmediatamente y pueden deshacerse con una sentencia rollback. Es importante tener en cuenta que, aunque las sentencias DDL se utilizan para crear, modificar y eliminar objetos de la base de datos, DML las sentencias se utilizan para manipular los datos dentro de esos objetos.

DML Las sentencias suelen ser ejecutadas por usuarios finales, como aplicaciones o sistemas que interactúan con la base de datos para recuperar, actualizar o eliminar datos. El sistema de gestión de bases de datos ejecuta DML sentencias después de realizar una consulta.

17.- Enliste los tipos de Usuarios de una BD.

Tipos de usuarios de bases de datos

Hay varios tipos de usuarios de bases de datos (DBMS), como:

- Administrador de base de datos (DBA)
- Usuario final
- Analista De Sistemas
- Programador de aplicaciones
- Diseñador de base de datos

18.- Defina Independencia de los Datos y sus tipos.

La independencia de los datos se refiere a la capacidad de realizar cambios en la estructura o el almacenamiento de los datos en una base de datos sin afectar las aplicaciones que acceden a esos datos. Hay dos tipos de independencia de datos:

Independencia física: Permite modificar la estructura de almacenamiento físico de los datos sin afectar las aplicaciones. Cambios en el almacenamiento, como la reorganización de archivos, deberían ser transparentes para las aplicaciones.

Independencia lógica: Permite modificar la estructura lógica de los datos (tablas, vistas, relaciones) sin afectar las aplicaciones que usan esos datos. Los cambios en la estructura de la base de datos deberían reflejarse en las aplicaciones sin requerir cambios significativos en el código.

La independencia de datos es un principio clave en los sistemas de gestión de bases de datos, ya que permite la adaptabilidad y evolución de la base de datos sin interrumpir las operaciones y aplicaciones existentes.

19.- ¿Qué es un Gestor de Archivos?

El gestor de archivos proporciona una manera sencilla e integrada de gestionar archivos y aplicaciones. Puede utilizar el gestor de archivos para realizar las siguientes acciones:

- Crear carpetas y documentos
- Mostrar archivos y carpetas
- Buscar y gestionar archivos
- Ejecutar secuencias de comandos e iniciar aplicaciones
- Personalizar la apariencia de archivos y carpetas
- Abrir ubicaciones especiales en el equipo
- Escribir datos en un CD o DVD
- Instalar y eliminar fuentes

El gestor de archivos gestiona el escritorio. El escritorio se encuentra detrás de todos los demás elementos visibles en la pantalla.

Todos los usuarios tienen una carpeta personal, que contiene todos los archivos relacionados con el usuario. El escritorio contiene iconos especiales que permiten un acceso sencillo a la papelera y la carpeta personal del usuario, y también a medios extraíbles, como disquetes, CD y unidades flash USB.

El gestor de archivos proporciona dos modos para interactuar con el sistema de archivos. Para elegir el modo que el gestor de archivos debe utilizar, seleccione Always Open In Browser Windows (Siempre abrir en una ventana de navegador) en la ficha Behavior (Comportamiento) del cuadro de diálogo de preferencias del gestor de archivos o anule la selección de esta opción. Los dos modos son:

Modo de explorador: En este modo, puede explorar los archivos y carpetas. La ventana del gestor de archivos representa un explorador, que puede mostrar cualquier ubicación. Al abrir una carpeta, se actualiza la ventana actual del gestor de archivos para mostrar el contenido de la nueva carpeta. La ventana del explorador muestra una barra de herramientas con acciones y ubicaciones frecuentes, una barra de direcciones que muestra la ubicación actual en la jerarquía de carpetas y una barra lateral que puede mostrar diferentes tipos de información. Para obtener más información sobre cómo usar el modo de explorador, consulte Modo de explorador.

Modo espacial: En este modo, puede navegar a los archivos y carpetas como objetos. La ventana del gestor de archivos representa una carpeta específica. Al abrir una carpeta, se abre una nueva ventana para esa carpeta. Cada vez que se abre una carpeta determinada, ésta se muestra en el mismo lugar de la pantalla y con el mismo tamaño con que se abrió la última vez. En el modo espacial, el gestor de archivos indica una carpeta abierta con un icono diferente. Usar el modo espacial puede hacer que haya más ventanas del gestor de archivos abiertas en la pantalla. Para obtener más información sobre cómo usar el modo espacial, consulte Modo espacial.

20.- ¿Qué hace un Procesador de Consultas?

El procesador de consultas y cálculo por defecto permite realizar el análisis en tiempo real mediante cálculos de procedimiento y capacidades de modelado de lectura y escritura.

El procesador de consultas y cálculo por defecto es el modo de agregación híbrida. El valor de configuración ASODYNAMICAGGINBSO controla si las bases de datos de almacenamiento de bloques utilizan el modo de agregación híbrida.

21.- Defina el Modelo E-R y especifique un poco de su historia, quien lo propuso, en que año.

El Modelo Entidad-Relación (E-R) es un enfoque conceptual utilizado en la gestión de bases de datos para diseñar y representar la estructura y las relaciones entre los datos de manera gráfica y comprensible. Fue propuesto para modelar de manera abstracta y visual las entidades (objetos o conceptos del mundo real), las relaciones entre ellas y los atributos que describen sus propiedades. Este modelo ayuda a los diseñadores de bases de datos a comprender y comunicar de manera efectiva la estructura y la semántica de una base de datos.

El modelo E-R fue desarrollado por Peter Chen, un científico de la computación nacido en China, en la década de 1970. El trabajo pionero de Chen en este campo se presentó en un artículo titulado "The Entity-Relationship Model: Toward a Unified View of Data" (El Modelo Entidad-Relación: Hacia una Visión Unificada de los Datos), publicado en la revista "ACM Transactions on Database Systems" en 1976.

El modelo E-R se convirtió rápidamente en un enfoque popular y ampliamente utilizado para el diseño conceptual de bases de datos. Facilitó la comunicación entre diseñadores de bases de datos y usuarios finales, ya que presentaba de manera clara y concisa cómo se relacionaban los datos en el mundo real. Con el tiempo, el modelo E-R se ha enriquecido con extensiones y refinamientos para abordar diversos aspectos y complejidades de la representación de datos en bases de datos.

22.- ¿Qué es Entidad?

Conceptualmente, una entidad es como una tabla de base de datos, y los atributos de entidad se corresponden con las columnas de la tabla. Crear un registro de entidad (o registro) en Dynamics 365 Customer Engagement (on-premises) es como agregar una fila a una tabla de base de datos. Las entidades se dividen en tres categorías: sistema, negocios y personalizadas. Como desarrollador que trabaja con datos profesionales, usará entidades de negocios y personalizadas. Dynamics 365 Customer Engagement (on-premises) usa las entidades del sistema para administrar todos los procesos internos, como los flujos de trabajo y los trabajos asincrónicos. No puede eliminar ni personalizar las entidades del sistema.

23.- Defina Atributo.

Un atributo es un dato que describe una entidad y existen los:

- a) Atributos simples:** Son atributos que no pueden dividirse en partes más pequeñas y representan una propiedad básica. Ejemplo: "Edad" de una persona.
- b) Atributos compuestos:** Son atributos que pueden dividirse en sub-atributos con significados propios. Ejemplo: "Dirección" que puede incluir "Calle", "Ciudad" y "Código Postal".
- c) Atributos de un solo valor:** Son atributos que contienen un único valor. Ejemplo: "Nombre" de una persona.
- d) Atributos de valor múltiple:** Son atributos que pueden contener varios valores. Ejemplo: "Teléfonos" de una persona, que podría tener varios números.
- e) Atributos derivados:** Son atributos cuyos valores se calculan a partir de otros atributos. No se almacenan directamente, sino que se derivan cuando es necesario. Ejemplo: "Edad" derivada a partir de la fecha de nacimiento y la fecha actual.

24.- ¿Qué es un conjunto de Entidades?

Un conjunto de entidades es un contenedor lógico para las instancias de un tipo de entidad y las instancias de cualquier tipo que se deriven de ese tipo de entidad. (Para obtener información sobre los tipos derivados, vea Entity Data Model: herencia). La relación entre un tipo de entidad y un conjunto de entidades es análoga a la relación entre una fila y una tabla en una base de datos relacional: al igual que una fila, un tipo de entidad describe la estructura de los datos y, al igual que una tabla, un conjunto de entidades contiene instancias de una determinada estructura. Un conjunto de entidades no es una construcción de modelado de datos, ya que no describe la estructura de los datos. En su lugar, un conjunto de entidades proporciona una construcción para que un entorno de hospedaje o de almacenamiento (como Common Language Runtime o una base de datos de SQL Server) agrupe las instancias del tipo de entidad y se puedan asignar a un almacén de datos.

Un conjunto de entidades se define dentro de un contenedor de entidades, que es una agrupación lógica de conjuntos de entidades y conjuntos de asociaciones.

Para que una instancia de tipo de entidad exista en un conjunto de entidades, deben cumplirse las condiciones siguientes:

- El tipo de la instancia puede ser el mismo que el tipo de entidad en el que se basa el conjunto de entidades, o un subtipo del tipo de entidad.
- La clave de entidad para la instancia es única dentro del conjunto de entidades.
- La instancia no existe en ningún otro conjunto de entidades.

25.- Defina lo que es una Relación.

Relaciones hace referencia a todas las relaciones que puede definir entre elementos del modelo (no sólo clases); por ejemplo, asociaciones, dependencias, generalización, flujos y enlaces. (En versiones anteriores de Rhapsody, el término hacía referencia a todos los diferentes tipos de asociaciones.)

26.- Describa brevemente los tipos de Relaciones que hay (Cardinalidad).

Define los atributos numéricos de la relación entre dos entidades o conjuntos de entidades. Las tres relaciones cardinales principales son uno a uno, uno a muchos, muchos a muchos y muchos a uno. Un ejemplo de uno a uno sería un estudiante asociado a una dirección de correo electrónico. Un ejemplo de uno a muchos (o muchos a uno, en función de la dirección de la relación) sería un estudiante que se inscribe en muchos cursos, y todos esos cursos se asocian a ese estudiante en particular. Un ejemplo de muchos a muchos sería los estudiantes en grupo están asociados a múltiples miembros de la facultad y a su vez los miembros de la facultad están asociados a múltiples estudiantes.

27.- ¿Qué es Cardinalidad de Mapeo?

La cardinalidad de mapeo es aquella mediante la cual puede especificarse la cantidad de entidades que podrán asociarse mediante una relación. La cardinalidad expresa cuántas entidades del conjunto de entidades de un extremo de la relación están relacionadas con cuántas entidades del conjunto del otro extremo. A esta cardinalidad también se le conoce como razón de cardinalidad. Por ejemplo, un artículo puede ser escrito por un solo autor o por varios, pero nunca por ninguno; un autor puede pertenecer a exactamente una institución (no para cero o varias); un artículo puede tener cero, uno o varios experimentos. Finalmente, un autor puede escribir muchos artículos, o ninguno.

28.- Defina los siguientes conceptos: Entidad Dominante, Entidad Subordinada y las relaciones existentes entre ambas.

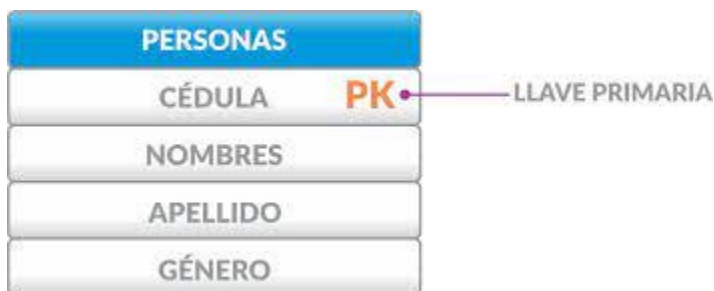
Entidad Dominante: En una relación entre dos entidades, la entidad dominante es aquella que tiene el control y contiene la clave principal. Suele ser la entidad principal de la relación y tiene la capacidad de existir independientemente. Por ejemplo, en una relación "Cliente tiene Ordenes", "Cliente" sería la entidad dominante, ya que las órdenes están vinculadas a los clientes.

Entidad Subordinada: En una relación entre dos entidades, la entidad subordinada es aquella que depende de la entidad dominante y no puede existir por sí misma en el contexto de esa relación. Por ejemplo, en la relación "Cliente tiene Ordenes", "Órdenes" sería la entidad subordinada, ya que las órdenes están relacionadas con los clientes, pero no tienen sentido independiente de ellos.

Las relaciones entre estas entidades generalmente se establecen mediante claves foráneas en la entidad subordinada que hacen referencia a la clave principal de la entidad dominante. Esto permite establecer conexiones y representar las asociaciones entre diferentes elementos en una base de datos u otro sistema de gestión de información.

29.- ¿Qué es una Llave Primaria?

Una clave principal es un campo o un conjunto de campos con valores únicos en una tabla. Los valores de la clave pueden usarse para hacer referencia a registros completos, porque cada registro tiene un valor diferente para la clave. Cada tabla solo puede tener una clave principal, básicamente es el Id único de la tabla.



30.- ¿Qué es una Llave Candidato?

Las claves candidatas son ese atributo o atributos que podrían servir como llaves primarias. Una llave candidata debe cumplir dos condiciones primordialmente:

- **Unicidad:** no pueden existir dos registros con el mismo valor en los atributos que forman la llave candidata.
- **Irreductibilidad o Minimalidad:** no existe ningún otro subconjunto de la llave que cumpla la regla de unicidad. En el caso de una llave candidata a la cual todavía se le puedan quitar campos y siga dándole unicidad al registro se deberá de hacer; es decir, si existe un conjunto de 4 campos que le dan unicidad al registro pero que se se le quita uno de esos campos (quedando en 3) aún sigue proporcionando la unicidad, entonces se debe optar por la llave de 3, y no la de 4 campos.

31.- ¿Qué es una Súper-llave?

Una Súper-llave es un concepto utilizado en bases de datos para referirse a un conjunto mínimo de atributos que puede identificar de manera única a cada registro en una tabla. En otras palabras, es un conjunto de uno o más atributos que, combinados, garantizan que no habrá registros duplicados en una tabla y que cada registro puede ser identificado de manera única. Esto es fundamental para asegurar la integridad y la eficiencia en la gestión de datos en bases de datos relacionales, básicamente es la combinación de dos atributos.

32.- Defina Entidad Fuerte.

Una Entidad Fuerte, en el contexto de las bases de datos, se refiere a una entidad que puede existir de manera independiente y tiene una identidad propia. En otras palabras, una entidad fuerte es aquella que tiene atributos únicos que la distinguen de otras entidades y puede ser representada por sí misma en una base de datos.

Por ejemplo, en una base de datos de una biblioteca, los libros podrían considerarse entidades fuertes. Cada libro tiene su propio conjunto de atributos únicos como título, autor, ISBN, etc., lo que permite que cada libro sea identificado y gestionado de manera individual en la base de datos.

33.- Defina Entidad Débil.

Una Entidad Débil, en el contexto de las bases de datos, es una entidad que depende de otra entidad para poder existir y ser identificada de manera única. En otras palabras, una entidad débil no tiene una identidad propia sin la presencia de una entidad fuerte relacionada.

Para representar una entidad débil en una base de datos, se requiere la existencia de una relación con una entidad fuerte, conocida como "entidad propietaria". La entidad débil no puede existir sin esta relación y depende de los atributos de la entidad propietaria para su identificación.

Por ejemplo, en una base de datos de gestión de estudiantes y cursos, la entidad "Inscripción" podría ser una entidad débil. No tendría identidad propia sin estar vinculada a las entidades "Estudiante" y "Curso", que actuarían como entidades propietarias y proporcionarían los atributos necesarios para identificar una inscripción de manera única.

34.- ¿Qué es la Generalización y que la Especialización en el modelo E-R. Herencia?

En el modelo Entidad-Relación (E-R), la Generalización se refiere al proceso de combinar varias entidades relacionadas en una entidad más general. La Generalización busca identificar características y atributos comunes entre estas entidades y crear una entidad genérica que englobe esas similitudes. Por otro lado, la Especialización es el proceso inverso, donde una entidad general se descompone en entidades más específicas que heredan atributos y características de la entidad original.

La Herencia en el contexto del modelo E-R se relaciona con la idea de que las entidades especializadas heredan atributos y características de una entidad general, permitiendo que las propiedades comunes se compartan entre ellas. Esto facilita la representación y el diseño de la base de datos al evitar la duplicación innecesaria de información y reflejar relaciones jerárquicas o de tipo "es un" entre las entidades.

35.- Especifique la simbología del modelo Entidad Relación.

Entidad (Rectángulo): Representa una clase de objetos con atributos propios. Se coloca el nombre de la entidad dentro del rectángulo.

Atributo (Ovalo): Representa una propiedad o característica de una entidad. Se conecta con líneas al rectángulo de la entidad.

Relación (Rombo): Representa la asociación entre dos o más entidades. El nombre de la relación se coloca en el rombo.

Atributo Clave (Subrayado): Un atributo subrayado en una entidad indica que es parte de la clave primaria de esa entidad.

Clave Primaria (Subrayado Doble): Un atributo subrayado doble en una entidad indica que es la clave primaria, es decir, el atributo o conjunto de atributos que identifica de manera única cada instancia de la entidad.

Línea de Relación: Une las entidades con la relación, indicando la conexión entre ellas.

Cardinalidad: Se representa usando números en los extremos de las líneas de relación para indicar cuántas instancias de una entidad están relacionadas con cuántas instancias de la otra entidad. Por ejemplo, "1" y "N" indican una relación uno a muchos.

Participación: Se puede indicar con símbolos "total" y "parcial" en los extremos de las líneas de relación para mostrar si todas las entidades de una clase deben participar en la relación (total) o si solo algunas pueden (parcial).

36.- Como se determina el grado de una relación?

El grado de una relación en el modelo Entidad-Relación (E-R) se refiere a la cantidad de entidades participantes en esa relación. Se clasifica en tres tipos principales:

Relación de grado uno (binaria): En este tipo de relación, participan dos entidades. Es la forma más común de relación y se representa mediante un rombo conectado por líneas a los rectángulos de las entidades involucradas.

Relación de grado dos (ternaria): En este tipo de relación, participan tres entidades. Se representa mediante un rombo conectado por líneas a tres rectángulos de entidades.

Relación de grado n (n-aria): En este tipo de relación, participan más de tres entidades. La cantidad exacta de entidades involucradas se indica mediante el número "n". Se representaría con un rombo conectado por líneas a n rectángulos de entidades.

El grado de una relación es importante para comprender la complejidad y las conexiones entre las entidades en una base de datos. Determinar el grado de una relación implica identificar cuántas entidades diferentes están involucradas en esa relación específica.

37.- Que es limitantes de mapeo?

Los limitantes de mapeo son reglas que deben de cumplir los datos contenidos en una base y nos permiten definir el número de entidades con las que se pueden relacionar otras entidades.

38.- Dependencia en existencia, relaciones exclusivas, restricción de exclusión, relaciones inclusivas (Tipos de Generalización).

Los conceptos que mencionaste están relacionados con los tipos de generalización en el modelo Entidad-Relación (E-R) y cómo se manejan las relaciones entre las entidades involucradas en un esquema de herencia.

Dependencia en Existencia: Se refiere a si una entidad débil puede existir por sí misma o si siempre debe estar relacionada con una entidad fuerte. Si la entidad débil no puede existir sin la entidad fuerte a la que está relacionada, hay una dependencia en existencia. Esta relación asegura que cada entidad débil tenga una entidad fuerte correspondiente.

Relaciones Exclusivas: En una jerarquía de generalización, las relaciones exclusivas indican que una entidad de la subclase solo puede ser parte de una superclase. En otras palabras, cada instancia de una subclase solo se relaciona con una instancia en la superclase.

Restricción de Exclusión: Similar a las relaciones exclusivas, pero aquí se establece que una entidad de la subclase puede estar asociada solo con una superclase específica y no con otras superclases.

Relaciones Inclusivas: Indican que una entidad de la subclase puede estar asociada con más de una superclase. Esto permite que una instancia de subclase esté relacionada con múltiples superclases en una jerarquía de generalización.

Estos conceptos son importantes para definir cómo las entidades se organizan y se relacionan dentro de una jerarquía de generalización, lo que afecta la forma en que se modela la herencia y las interacciones entre las entidades en una base de datos.

Referencias:

<https://www.ionos.es/digitalguide/servidores/know-how/modelo-cliente-servidor/#:~:text=En%20el%20modelo%20cliente%2Dservidor,inicia%20las%20tareas%20del%20servidor.>

<https://www.ibm.com/docs/es/aix/7.1?topic=systems-client-server>

<https://www.guiaspracticas.com/ordenadores-de-sobremesa/computadora-de-red#:~:text=Podemos%20distinguir%20varios%20tipos%20de,local%2C%20S%C3%AD%20usa%20procesamiento%20local.>

<https://www.claranet.es/blog/que-tipos-de-servidores-hay>

<https://www.netec.com/para-que-sirve-una-base-de-datos#:~:text=Una%20base%20de%20datos%20permite,y%20simple%20desde%20un%20ordenador.&text=Fila%3A%20cada%20una%20conforma%20un%20registro.>

[https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos#:~:text=Un%20sistema%20gestor%20de%20base%20de%20datos%20\(SGBD\)%20o%20Database, incluye%20una%20base%20de%20datos.](https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos#:~:text=Un%20sistema%20gestor%20de%20base%20de%20datos%20(SGBD)%20o%20Database, incluye%20una%20base%20de%20datos.)

<https://www.stackscale.com/es/blog/sistemas-administracion-bases-datos-populares/>

<https://appmaster.io/es/blog/diferencia-entre-ddl-y-dml>

<https://learn.microsoft.com/es-es/sql/database-engine/configure-windows/database-engine-instances-sql-server?view=sql-server-ver16>

<https://www.ibm.com/mx-es/topics/database-schema#:~:text=Un%20esquema%20de%20base%20de%20datos%20define%20c%C3%B3mo%20se%20organizan,las%20relaciones%20entre%20estas%20entidades.>

<https://appmaster.io/es/blog/diferencia-entre-ddl-y-dml>

https://docs.oracle.com/cd/E37929_01/html/E36713/gldhq.html

<http://dbadixit.com/llaves-candidatas-claves-candidatas/#:~:text=Las%20claves%20candidatas%20son%20ese,que%20forman%20la%20llave%20candidata.>

<https://learn.microsoft.com/es-es/dynamics365/customerengagement/on-premises/developer/introduction-entities?view=op-9-1>

<https://www.ibm.com/docs/es/engineering-lifecycle-management-suite/design-rhapsody/9.0.1?topic=elements-defining-relations>

<https://www.lucidchart.com/pages/es/que-es-un-diagrama-entidad-relacion#:~:text=m%C3%BAltiples%20valores%20compuestos,-,Cardinalidad,una%20direcci%C3%B3n%20de%20correo%20electr%C3%B3nico.>

<https://www.coursehero.com/file/66437447/Cardinalidad-de-Mapeo/docx/>

<https://support.microsoft.com/es-es/office/agregar-o-cambiar-la-clave-principal-de-una-tabla-en-access-07b4a84b-0063-4d56-8b00-65f2975e4379#:~:text=Una%20clave%20principal%20es%20un,puede%20tener%20una%20clave%20principal.>

https://prezi.com/n_alhnpke781/limitantes-de-mapeo-diagrama-entidad-relacion/#:~:text=CONCLUSION%3A%20Los%20limitantes%20de%20mapeo,identidades%20otravez%20de%20una%20relaion.