DATABASE

Alumna: Natalí Denise Rey
Prof. : Camilo Andrés Redondo
Tutor: Jorge Rodríguez
Inst.: CoderHouse

Índice

Introducción

La base de datos que se presenta a continuación ha sido desarrollada para gestionar la información de un gimnasio, buscando facilitar la administración de los distintos aspectos relevantes para el funcionamiento eficiente del establecimiento.

Objetivo

El objetivo principal de esta base de datos es proporcionar una herramienta amigable y eficaz que permita a los administradores del gimnasio realizar un seguimiento detallado de la información clave. Busca mejorar la organización y accesibilidad de los datos.

Problemática

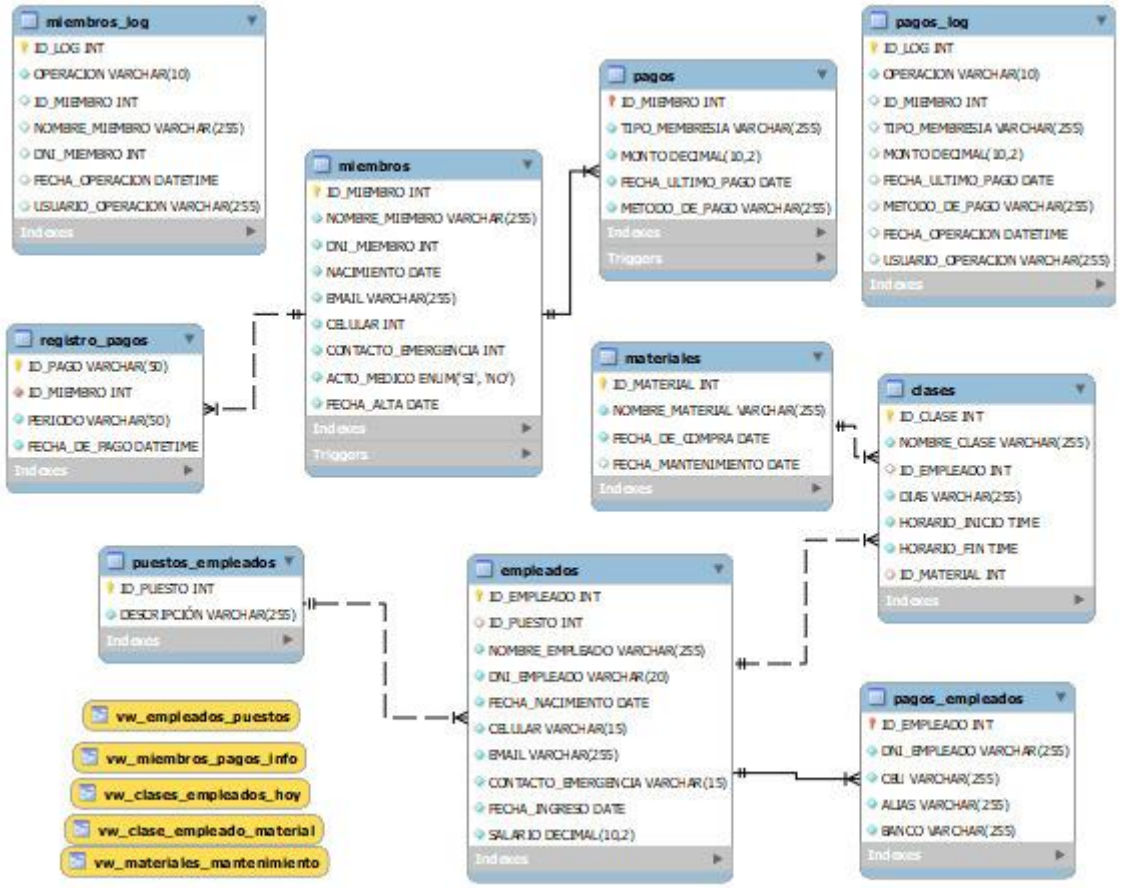
Antes de la implementación de esta base de datos, el gimnasio podría haber enfrentado desafíos relacionados con la dificultad para mantener un registro preciso de los miembros, realizar un seguimiento de los pagos, gestionar las clases y supervisar la información del personal. Estos problemas podrían haber llevado a una experiencia menos satisfactoria tanto para los miembros como para el personal administrativo. Con el crecimiento de caudal de miembros y por lo tanto, de personal, es necesario contar con una herramienta más avanzada y eficiente.

Herramientas utilizadas

- ➔ Workbench 8.0
- ➔ GitHub
- ➔ Excel (reporte)

DER : Diagrama - Entidad - Relación

Ayuda a diseñar y visualizar cómo la información se organiza y se relaciona en una base de datos.



Tablas

Las tablas almacenan la información organizada.
A continuación detallaremos las tablas que componen nuestra database (DB) y describiremos los campos que las componen:

Tabla **MIEMBROS**

En ella almacenaremos la información de los miembros del gimnasio. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_MIEMBRO: Este campo almacena un número entero. El ID (identificador) será el "DNI" dentro de nuestra DB de cada miembro.

NOMBRE_MIEMBRO: Este campo almacena cadena de caracteres, es decir, texto. Allí podremos ingresar el nombre completo de los miembros.

DNI_MIEMBRO: Este campo almacena un número entero, en este caso el DNI del miembro.

NACIMIENTO: Este campo almacena fechas. Acá debemos ingresar la fecha de nacimiento del miembro.

EMAIL: Este campo almacena texto y debemos ingresar el correo electrónico del miembro.

CELULAR: Este campo almacena un número entero, en este caso el celular del miembro.

CONTACTO_EMERGENCIA: Este campo almacena un número entero, en este caso un contacto de emergencia por si el miembro tiene algún problema.

ACTO_MEDICO: Este dato está definido con dos tipos de valores: si o no. Se utilizará la opción que corresponda. Todos los miembros deben presentar el acto médico.

FECHA_ALTA: Este campo almacena fechas. Acá debemos ingresar la fecha de alta en el gimnasio.

Tabla **PAGOS**

En ella almacenaremos la información de los pagos del gimnasio. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_MIEMBRO: Este ID es el mismo que cargamos en la tabla miembros.

TIPO_MEMBRESIA: Este campo almacena texto, allí agregaremos el tipo de membresía que el miembro optó

MONTO: En este campo almacenaremos el monto mensual o anual según corresponda del costo de la membresía. Se podrá ingresar un número decimal de hasta 10 dígitos y 2 dígitos a la derecha del punto decimal.

FECHA_ULTIMO_PAGO: Este campo almacena fechas. Acá debemos ingresar la fecha de último pago del miembro.

METODO_DE_PAGO: Este campo almacena texto, agregaremos el método de pago que el miembro optó para abonar su membresía.

Tabla **REGISTRO_PAGOS**

En ella almacenaremos el registro de todos los pagos que los miembros realizaron. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_PAGO: Este campo almacena texto. El ID (identificador) dentro de nuestra DB de cada pago.

ID_MIEMBRO: Este ID es el mismo que cargamos en la tabla miembros y es un número entero.

PERIODO: Este campo almacenará texto: el período de la factura que se abonó, por ejemplo: diciembre 2023.

FECHA_DE_PAGO: Este campo almacena fecha y hora exacta en que se abonó la factura.

Tabla **MATERIALES**

En ella almacenaremos el registro de los materiales que tiene el gimnasio para poder llevar un seguimiento de mantenimiento de los mismos y para organizar en sus respectivas clases. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_MATERIAL: Este campo almacena un número entero. El ID (identificador) será el "DNI" dentro de nuestra DB de cada material.

NOMBRE_MATERIAL: Este campo almacena texto. Allí podremos ingresar el nombre del material.

FECHA_DE_COMPRA: Este campo almacena fechas. Acá debemos ingresar la fecha de compra del material.

FECHA_MANTENIMIENTO: Este campo almacena fechas. Acá debemos ingresar la fecha estipulada de mantenimiento del material.

Tabla **PUESTOS_EMPLEADOS**

En ella almacenaremos la descripción de los distintos puestos que desempeñan los empleados en nuestro gimnasio. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_PUESTO: Este campo almacena un número entero. El ID (identificador) será el "DNI" dentro de nuestra DB de cada puesto.

DESCRIPCIÓN: Este campo almacena texto. Describe los puestos que representa cada identificador de puesto.

Tabla **EMPLEADOS**

En ella almacenaremos la información más relevante de los empleados del gimnasio. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_EMPLEADO: Este campo almacena un número entero. El ID (identificador) será el "DNI" dentro de nuestra DB de cada empleado.

ID_PUESTO: Este campo almacena un número entero. El ID (identificador) del puesto que desempeña el empleado.

NOMBRE_EMPLEADO: Este campo almacena cadena de caracteres, es decir, texto. Allí podremos ingresar el nombre completo de los empleados.

DNI_MIEMBRO: Este campo almacena un número entero, en este caso el DNI del empleado.

FECHA_NACIMIENTO: Este campo almacena fechas. Acá debemos ingresar la fecha de nacimiento del empleado.

CELULAR: Este campo almacena un número entero, en este caso el celular del empleado.

EMAIL: Este campo almacena texto y debemos ingresar el correo electrónico del empleado.

CONTACTO_EMERGENCIA: Este campo almacena un número entero, en este caso un contacto de emergencia por si el empleado tiene algún problema.

FECHA_INGRESO: Este campo almacena fechas. Acá debemos ingresar la fecha que el empleado comenzó a trabajar.

SALARIO: Este campo almacena el salario del empleado. Se podrá ingresar un número decimal de hasta 10 dígitos y 2 dígitos a la derecha del punto decimal.

Tabla **CLASES**

En ella almacenaremos la información sobre las clases que se dictan en el gimnasio. Todos los campos deben contener información, nunca pueden estar vacíos.

ID_CLASE: Este campo almacena un número entero. El ID (identificador) será el "DNI" dentro de nuestra DB de cada clase.

NOMBRE_CLASE: Este campo almacena cadena de caracteres, es decir, texto. Allí podremos ingresar el nombre de la clase.

ID_EMPLEADO: Este campo almacena un número entero. Definimos el empleado que dictará la clase según el mismo identificador que se le otorgó en la tabla EMPLEADOS.

DIAS: Este campo almacena texto, allí ingresamos los días en que se dicta la clase.

HORARIO_INICIO: Este campo almacena un horario. Ingresamos el horario en que inicia la clase.

HORARIO_FIN: Este campo almacena un horario. Ingresamos el horario en que finaliza la clase.

ID_MATERIAL: Este campo almacena un número entero. Ingresamos el identificador del material que se necesitará ocupar en la clase. Este ID lo sacamos de la tabla MATERIALES.

Tabla **PAGOS_EMPLEADOS**

En ella almacenaremos los datos bancarios de nuestros empleados.

ID_EMPLEADO: Este campo almacena un número entero. El identificador de nuestro empleado de la tabla EMPLEADOS.

NOMBRE_EMPLEADO: Este campo almacena texto. Tendremos el nombre del empleado de la tabla EMPLEADOS.

CBU: Este campo almacena cadena de carácteres, allí guardaremos el CBU del empleado.

ALIAS: Este campo almacena texto. Guardaremos el alias bancario del empleado.

BANCO:Este campo almacena texto. Guardaremos el nombre del banco del empleado.

Tablas de auditoría

Estas tablas son conjuntos de datos adicionales que se crean automáticamente en respuesta a ciertos eventos en la base de datos y tienen el propósito de mantener un historial detallado de las acciones realizadas en una base de datos, lo

que permite realizar un seguimiento de quién hizo qué y cuándo.

Tabla **MIEMBROS_LOG**

Esta tabla se genera antes y después de insertar un nuevo miembro en la tabla MIEMBROS. Deja un registro de la fecha, usuario y miembro que se insertó

ID_LOG: Este campo almacena un número entero y es un identificador único que se genera para el registro de cada operación.

OPERACION: Este campo almacena texto y describe el tipo de operación que se realizó.

ID_MIEMBRO: Este campo almacena un número entero. En este caso el identificador del nuevo miembro ingresado.

NOMBRE_MIEMBRO: Este campo almacena texto. Allí podremos ingresar el nombre completo de los miembros.
DNI_MIEMBRO: Este campo almacena un número entero. Será el DNI del miembro ingresado.

FECHA_OPERACION: Es un campo que almacena fecha y horario en que se realizó la operación.

USUARIO_OPERACION: Describe quién realizó la operación y es un campo que almacena texto.

Tabla **PAGOS_LOG**

Esta tabla se genera antes y después de modificar un dato importante en la tabla PAGOS. Deja un registro de la fecha, usuario y modificación que se realizó.

ID_LOG: Este campo almacena un número entero y es un identificador único que se genera para el registro de cada operación.

OPERACION: Este campo almacena texto y describe el tipo de operación que se realizó.

ID_MIEMBRO: Este identificador será del miembro al que se realizó alguna modificación en un dato importante de la tabla PAGOS. (Número entero establecido desde el registro del miembro en la tabla MIEMBROS)

TIPO_MEMBRESIA: Este campo almacena texto y describe el tipo de membresía que el miembro optó o modificó.

MONTO: Este campo almacena un número entero y describe el monto que le corresponde al miembro.

FECHA_ULTIMO_PAGO: Este campo almacena una fecha. Allí encontraremos la fecha de último pago de la membresía.

METODO_DE_PAGO: Este campo almacena texto y describe el método de pago que el miembro optó o modificó.
FECHA_OPERACION: Es un campo que almacena fecha y horario en que se realizó la operación.

USUARIO_OPERACION: Describe quién realizó la operación y es un campo que almacena texto.

Vistas

Las vistas son consultas [1] predefinidas que te permiten ver datos almacenados en una o más tablas de una manera específica y, sobre todo, de una manera mucho más sencilla para aquellos usuarios que no conocen la estructura interna de la relación entre tablas.

1. Consultas: Solicitud que se realiza a una base de datos para recuperar, modificar o manipular datos. Son esenciales para interactuar con bases de datos y obtener información específica de las tablas que las componen.

Vista **vw_miembros_pagos_info**

Esta vista fusiona la tabla MIEMBROS , PAGOS y REGISTRO_PAGOS y nos trae la información más relevante sobre los pagos de los miembros.

Vista **vw_empleados_puestos**

Esta vista fusiona la tabla EMPLEADOS y PUESTOS_EMPLEADOS para que al ejecutarla nos de como resultado todos los empleados de nuestro gimnasio y con la descripción de su respectivo puesto.

Vista **vw_clases_empleados_hoy**

Esta vista fusiona la tabla CLASES Y EMPLEADOS y tiene como parámetro el día actual. Al ejecutarla nos traerá la información de las clases que se dictarán en el día que se ejecuta la vista y también traerá el nombre de los empleados que se harán cargo de las mismas.

Vista **vw_clase_empleado_material**

Esta vista fusiona la tabla CLASES, EMPLEADOS y MATERIALES. Nos traerá el nombre del empleado, el material que utilizará y el nombre de la clase en la que lo utilizará.

Vista **vw_materiales_mantenimiento**

Esta vista utiliza la tabla MATERIALES y utiliza como parámetro la fecha actual. Esta vista nos traerá la lista de materiales que la fecha de mantenimiento sea anterior a la actual. Es decir, traerá los materiales que necesitan mantenimiento.

Funciones

Las funciones permiten procesar y manipular los datos de forma eficiente. Cada función se crea con un objetivo específico que nos será de utilidad.

Función **calcularPrecioTotal**

Esta función nos permite realizar un cálculo. En este caso, es para obtener cuánto quedaría el monto a cobrar al miembro si se le aplica un descuento. El usuario podrá ingresar como parámetro el valor de la cuota actual y el porcentaje de descuento a aplicar y nuestra función traerá ese resultado.

Función **calcularAntiguedadPorDNI**

Esta función nos permite obtener la antigüedad de los miembros utilizando el DNI como parámetro.

Triggers

Los triggers o "disparadores" se ejecutan automáticamente en respuesta a ciertos eventos en la DB. Cuando detecta ciertos eventos, activa automáticamente un conjunto de acciones predefinidas. Utilizamos los triggers para automatizar el registro de ciertos eventos en las tablas de log (tablas de auditoría).

Trigger **before_insert_miembros**

Este trigger está diseñado para ejecutarse antes de realizar una operación de inserción de datos en la tabla MIEMBROS. Su propósito es verificar si el valor del nuevo DNI que se está intentando insertar ya existe en la tabla. Si encuentra una coincidencia, emite una señal de error con un mensaje específico.

Trigger **after_insert_miembros**

Este trigger se ejecuta después de que se realiza una operación de inserción de datos en la tabla MIEMBROS. Su

función principal es registrar la operación de inserción en una tabla de registro llamada MIEMBROS_LOG.

Trigger **before_update_pagos**

Este trigger se ejecuta antes de realizar una operación de actualización en la tabla PAGOS. Su función principal es verificar si hay cambios significativos en ciertos campos específicos y, en caso afirmativo, registrarlo en una tabla de auditoría llamada PAGOS_LOG.

Trigger **after_update_pagos**

Este trigger se ejecuta después de que se realiza una operación de actualización en la tabla PAGOS.

Store procedure

Los Store procedure son un conjunto de instrucciones predefinidas y almacenadas en la base de datos. Estos procedimientos proporcionan una forma de encapsular lógica de negocio y operaciones complejas en la propia base de datos.

Store procedure **OrdenarTablaMiembros**
Sirve para ordenar la tabla MIEMBROS según una columna específica y en una dirección específica (ascendente o descendente).

Store procedure **ModificarFechaUltimoPago**
Sirve para modificar la columna FECHA_ULTIMO_PAGO en la tabla PAGOS para un miembro específico.

Store procedure **InsertarMiembro**
Sirve para insertar nuevos registros en la tabla MIEMBROS.

DCL (Data Control Language)

Se han ejecutado un conjunto de comandos utilizados para gestionar los permisos y la seguridad en una base de datos. Esto permite controlar el acceso a los datos y la realizaciónde operaciones sobre la base de datos, determinando quién tiene qué tipo de privilegios y qué acciones pueden realizar.

Para esta DB se ha creado un usuario para solo lectura de todas las tablas y otro usuario con permiso de lectura, inserción y modificación de los datos de todas las tablas. En ambos usuarios se prohibió la eliminación de registros.

TCL (Transaction Control Language)

Se han implementado un conjunto de comandos que se utilizan para realizar operaciones en una base de datos de manera controlada.

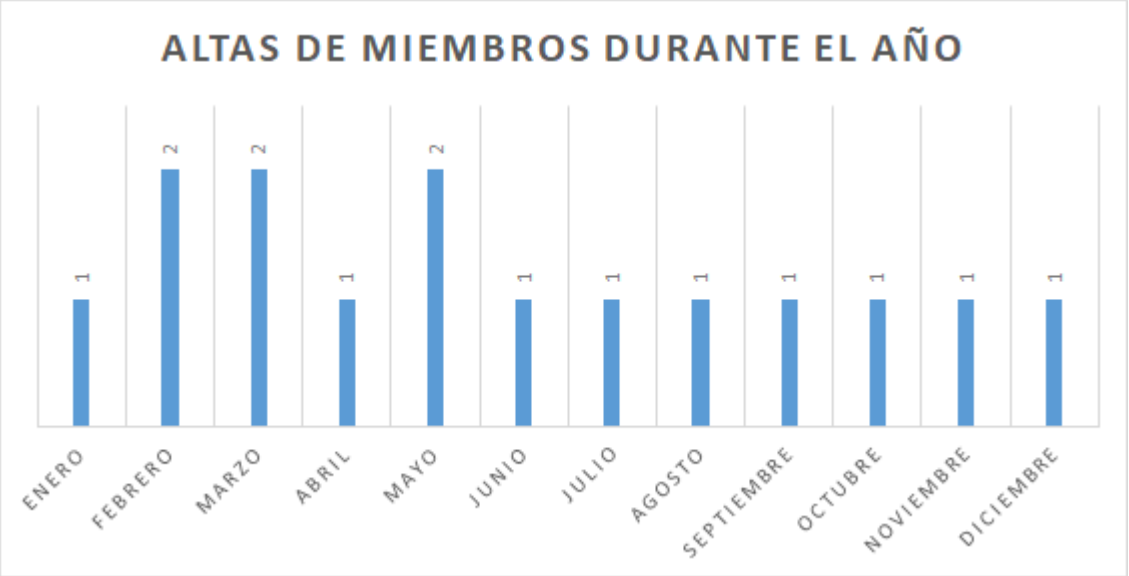
Backup

El backup de una base de datos es una copia de seguridad que se realiza para preservar la integridad y disponibilidad de la información almacenada en la base de datos. Se realizó el backup de toda la DB.

Reporte

Los reportes son presentaciones organizadas y resumidas de la información almacenada en la base de datos. Sirven para extraer significado y valor de los datos almacenados, ya que ofrecen una visión estructurada y clara de la información relevante.

En el siguiente reporte evaluaremos en qué meses los miembros se dan de alta con mayor frecuencia, este reporte nos servirá para determinar la "temporada alta" del gimnasio y a partir de él tomar decisiones que favorezcan al desarrollo eficiente del gimnasio.



Conclusión

En el transcurso del curso de SQL, tomé la decisión de desarrollar una base de datos para un gimnasio. Al reflexionar sobre esta elección, considero que fue acertada. Esta decisión me permitió evitar perder tiempo en la complejidad de la información que se insertaría y centrarme en aprender aspectos técnicos desde lo básico hasta lo avanzado. Además, posibilitó la aplicación práctica de los temas discutidos en clase, abordando cuestiones esenciales como la seguridad de la información y la integridad de los datos. Estos aspectos son cruciales para la gestión eficiente no solo de un gimnasio, sino de cualquier tipo de negocio.

El curso resultó muy enriquecedor, ya que actualmente aplico los conocimientos adquiridos al realizar consultas en bases de datos en mi trabajo actual. Este aprendizaje ha agilizado significativamente la obtención de información y ha mejorado mi comprensión del funcionamiento interno de las bases de datos.