

Proyecto final de SQL



⑩ Alumno: Pablo Burgos

⑩ Docente: Camilo Andrés Redondo

⑩ Curso: SQL

⑩ Comisión: 34960

Indice

| | |
|--|----|
| Introducción | 3 |
| Objetivo..... | 3 |
| Situación problemática..... | 3 |
| Diagrama entidad relación | 4 |
| Definición de tablas | 5 |
| Vistas..... | 8 |
| Funciones | 9 |
| Stored Procedures | 9 |
| Triggers | 10 |
| Transactions | 12 |
| Sentencias..... | 12 |
| Backup | 13 |
| Herramientas y metodologías usadas | 13 |

Introducción

El siguiente proyecto corresponde a que cualquier empresa chica o mediana pueda utilizar dicha base de datos para almacenar la información tanto de sus productos como de las transacciones.

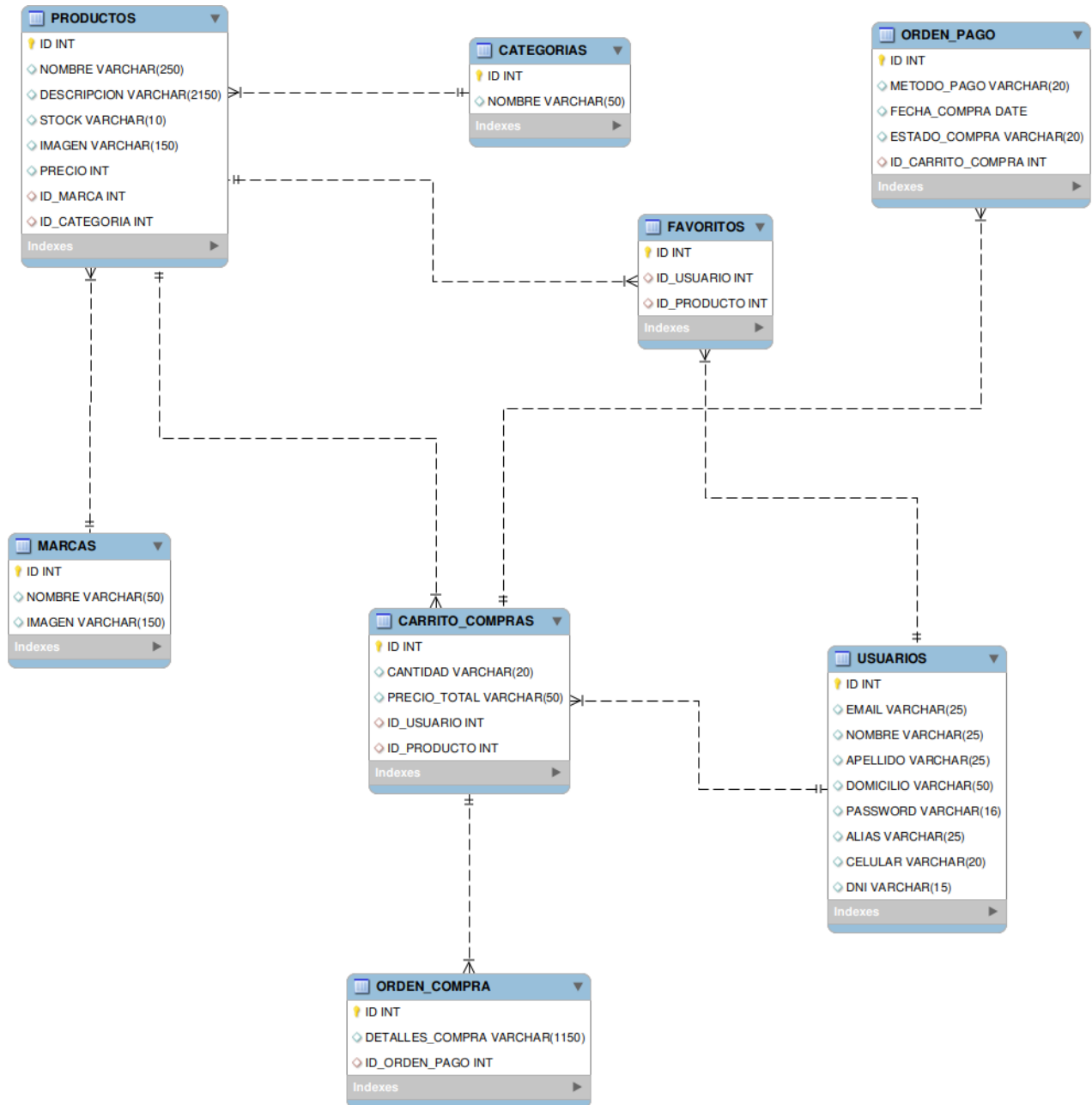
Objetivo

El objetivo es centralizar todos los datos que correspondan a un e-commerce, donde se pueda solicitar en una sola ubicación la información necesaria en cada caso, de esta manera generar eficiencia, orden y claridad en los pedidos y datos.

Situación problemática

La problemática inicial radica en eliminar completamente el uso de estanterías con archivos llenos de carpeta con hojas a veces escrito a mano otras impresas pero que al final algún siniestro podría dejar inutilizable dichos archivos. En segunda instancia aplicar la tecnología para centralizar y digitalizar dichos archivos seria ideal, ademas implementar el almacenamiento de todas las transacciones para tener un control tanto mensual, anual y total, poder aplicar métricas sacando directamente de los datos guardados.

Diagrama entidad relación



Definición de tablas

A continuación se describen las tablas con los datos detallados:

- ⑩ Usuarios: La tabla usuarios contiene la información de cada usuario que quiera realizar una compra en el e-commerce.

| Nombre del campo | Tipo de campo |
|---|---------------|
| ID(PK): Código único asignado a cada usuario | INT |
| EMAIL: Correo electrónico del usuario | VARCHAR(25) |
| NOMBRE: Nombre del usuario | VARCHAR(25) |
| APELLIDO: Apellido del usuario | VARCHAR(25) |
| DOMICILIO: Domicilio del usuario | VARCHAR(150) |
| PASSWORD: contraseña del usuario para acceder a su cuenta | VARCHAR(16) |
| ALIAS: sobrenombre del cliente | VARCHAR(25) |
| CELULAR: Celular del usuario | VARCHAR(20) |
| DNI: Documento de identidad del usuario | VARCHAR(15) |

- ⑩ Marcas: La tabla marcas contiene información de cada marca y puede estar asociado a algún producto en particular.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada marca | INT |
| NOMBRE: Nombre de la marca | VARCHAR(25) |
| IMAGEN: Contiene un link con la imagen de la marca | VARCHAR(150) |

- ⑩ Categorías: La tabla categorias contiene la información de las categorías que pueden estar asociadas a varios productos.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada categoría | INT |
| NOMBRE: Nombre de la marca | VARCHAR(50) |

- ⑩ Productos: La tabla productos contiene la información de cada producto que se vende en el e-commerce.

| Nombre del campo | Tipo de campo |
|---|---------------|
| ID(PK): Código único asignado a cada producto | INT |
| NOMBRE: Nombre del producto | VARCHAR(250) |
| DESCRIPCION: Detalles del producto | VARCHAR(2150) |
| STOCK: Cantidad disponible del producto | VARCHAR(10) |
| IMAGEN: Contiene un link con la imagen del producto | VARCHAR(150) |
| PRECIO: Costo del producto | INT |
| ID_MARCA(FK): Numero que identifica la marca del producto | INT |
| ID_CATEGORIA: Numero que identifica la categoría del producto | INT |

- ⑩ Favoritos: La tabla favoritos contiene la información de favoritos de cada usuario.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada favorito | INT |
| ID_USUARIO(FK): Numero de identifica a un usuario | INT |
| ID_PRODUCTO(FK): Numero que identifica un producto | INT |

- ⑩ Carrito de compras: La tabla carrito_compras contiene la información de cada carrito de compras asignado a un usuario en particular con un producto y cantidad.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada carrito de compras | INT |
| CANTIDAD: Indica la cantidad que quiere comprar un usuario | VARCHAR(20) |
| PRECIO_TOTAL: Suma total de la cantidad del producto | VARCHAR(50) |
| ID_USUARIO(FK): Numero que identifica un usuario asignado a este carrito | INT |
| ID_PRODUCTO(FK): Numero que identifica al producto añadido | INT |

- ⑩ Orden de pago: La tabla orden_pago contiene información sobre la orden de pago de un usuario que medio de pago usó, la fecha y el estado de dicha orden.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada orden de pago | INT |
| METODO_PAGO: Contiene la forma de pago del usuario | VARCHAR(20) |
| FECHA_COMPRA: Contiene la fecha que realizó la compra | DATE |
| ESTADO_COMPRA: Indica el estado de compra(procesado, no procesado, etc) | VARCHAR(20) |
| ID_CARRITO_COMPRA: Numero que identifica a un carrito de compras en particular | INT |

- ⑩ Orden de compra: La tabla orden_compra contiene la información de la orden de compra.

| Nombre del campo | Tipo de campo |
|------------------|---------------|
|------------------|---------------|

| | |
|---|---------------|
| ID(PK): Código único asignado a cada orden de compra | INT |
| DETALLES_COMPRA: contiene el detalle de la compra | VARCHAR(1150) |
| ID_ORDEN_PAGO: numero que indica la orden de pago asignada a cada orden de compra | INT |

Vistas

Se crean 5 vistas en base a información de las tablas, a continuación se detallan cada una de ellas:

1. favoritos: En esta vista muestra todos los favoritos por usuario.
2. comics: En esta vista muestra todos los productos con categoría cómics.
3. manga: En esta vista muestra todos los productos con categoría manga.
4. compras_procesadas: En esta vista muestra todas las ordenes de pago con estado de compra “procesado”.
5. productos_800up: En esta vista muestra todos los productos con un precio por encima de 800 pesos.

Funciones

Se crean 2 funciones en base a información de las tablas, a continuación se detallan las mismas:

1. cantidad_favoritos_usuarios: Esta función me da como resultado la cantidad total de favoritos del usuario que se pase como parámetro (recibe un valor VARCHAR(30)) .
2. total_carrito_compras: En esta función da como resultado el total de gasto sumando todos los carritos del usuario que se pase como parámetro (recibe un valor VARCHAR(30)) .

Stored Procedures

Se crean 2 stored procedures en base a la información de las tablas, a continuación se detallan las mismas:

1. sp_lista_ordenada: En este stored procedure da como resultado la lista ordenada ya sea de forma descendente o ascendente de cualquier columna de la tabla productos. Recibe dos parámetros, el primero la columna a ordenar (recibe un valor VARCHAR(20)) y el segundo si es ascendente o descendente (recibe un valor VARCHAR(20)).
2. sp_eliminar_usuario: En este stored procedure se elimina un usuario por el ID que se pasa como parámetro (recibe un valor INT).

Triggers

Se crean 4 triggers que se activan dependiendo de una acción que se realice en las tablas productos y/o usuarios. Inicialmente se crea una tabla “LOG_AUDITORIA” donde se almacenara todas las acciones que se realicen adjuntando los datos requeridos en cada trigger.

- ⑩ log_auditoria: Esta tabla se crea para almacenar los datos que se inserten o actualicen en las tablas de usuarios y/o productos, cada campo se va a llenar dependiendo del trigger.

| Nombre del campo | Tipo de campo |
|--|---------------|
| ID(PK): Código único asignado a cada acción | INT |
| EMAIL: Se guarda el email cuando se inserta o actualiza un usuario | VARCHAR(25) |
| PASSWORD: Se guarda el password cuando se inserta o actualiza un usuario | VARCHAR(16) |
| DNI: Se guarda el dni cuando se inserta o actualiza un usuario | VARCHAR(15) |
| NOMBRE: Se guarda el nombre cuando se inserta o actualiza un producto | VARCHAR(250) |
| STOCK: Se guarda el stock cuando se inserta o actualiza un producto | VARCHAR(10) |
| PRECIO: Se guarda el precio cuando se inserta o actualiza un producto | INT |
| CAMPOANTERIOR_CAMPONUEVO: Se guarda los valores antiguos y los nuevos valores cuando se hace un update de las tablas usuario y/o productos | VARCHAR(500) |
| NOMBRE_ACCION: Se guarda la acción que se realiza (insert o update) | VARCHAR(10) |
| NOMBRE_TABLA: Se guarda el nombre de la tabla donde se realiza alguna acción | VARCHAR(50) |
| USUARIO: Se guarda el usuario que hizo dicha acción | VARCHAR(100) |
| FECHA_UPD_INS_DEL: Se guarda la fecha en que se realizó la acción | DATETIME |

1. trg_log_productos_i: Este trigger se activa después de hacer un insert en la tabla productos, almacenando en “LOG_AUDITORIA” los campos nombre, stock, precio, la acción “insert”, la tabla en donde se inserta que en este caso es productos, el usuario que realizó el insert y la fecha/hora de la misma.

2. trg_log_productos_u: Este trigger se activa antes de hacer un update en la tabla de productos, almacenando en “LOG_AUDITORIA” los campos nuevos de nombre, stock, precio, en “campoanterior_camponuevo” se almacena los valores antiguos junto con los nuevos, la acción “update”, la tabla en donde se hace el update en este caso productos, el usuario que realizó el update y la fecha/hora de la misma.
3. trg_log_usuarios_i: Este trigger se activa después de hacer un insert en la tabla usuarios, almacenando en “LOG_AUDITORIA” los campos email, password, dni, la acción “insert”, la tabla en donde se inserta que en este caso es usuarios, el usuario que realizó el insert y la fecha/hora de la misma.
4. trg_log_usuarios_u: Este trigger se activa antes de hacer un update en la tabla de usuarios, almacenando en “LOG_AUDITORIA” los campos nuevos de email, password, dni, en “campoanterior_camponuevo” se almacena los valores antiguos junto con los nuevos valores, la acción “update”, la tabla en donde se hace el update en este caso usuarios, el usuario que realizó el update y la fecha/hora de la misma.

Transactions

Se crean 2 transaction, para ejecutar primero deshabilitamos la función autocommit, luego se indica el comienzo de una transacción con STAR TRANSACTION y sobre la tabla usuarios se elimina a tres usuarios mediante sus Ids, en este caso el 1, 3 y 5, se ejecuta la sentencia y se verifica que los usuarios se eliminaron. Agregando el comando ROLLBACK se puede revertir las acciones anteriores y volver al punto de STAR TRANSACTION. La segunda transacción se indica el comienzo con STAR

TRANSACTION y se procede a agregar cuatro ítems a la tabla productos, seguido se agrega comando SAVEPOINT, Siguiendo con el transaction se vuelve a agregar cuatro ítems mas a la tabla productos y se vuelve a agregar un SAVEPOINT. Este ultimo comando que se le nombra de alguna forma y actúa como un punto de salvedad en caso de que lo que esté siguiente a este savepoint no quisiéramos aplicarlo se aplica un RELEASE SAVEPOINT con el nombre de dicho savepoint y lo que se hizo después de dicho savepoint no se aplica.

Sentencias

Se crean 2 usuarios que tiene diferentes permisos con respecto a la base de datos, a continuación los detalles:

1. l_user: Se crea este usuario solamente con permisos de lectura, puede hacer queries con pedidos de información a la base de datos pero no puede editar, agregar, borrar ni editar.
2. lim_user: Se crea este usuario con permisos de lectura, modificación e inserción pero no borrar datos de la base de datos.

Backup

Se realiza backup de la base de datos completa, con las tablas y el contenido de cada una, se incluyen funciones, vistas, stored procedures y triggers.

Herramientas y metodologías usadas

MySQL Workbench

LibreOffice Writer

LibreOffice Calc