



PLATAFORMA DE ENTREGA

PROYECTO



2023-12-12

1-DAM_IES PERE MARIA ORTS I BOSCH
[Alumno: Yu Zhang]

ÍNDICE

REQUISITOS:.....	2
ESQUEMA DE ENTIDAD-RELACIÓN:.....	4
DISEÑO LÓGICO MODELO RELACIONAR:.....	5
LAS CONSULTAS REALIZADOS:	10
--a). 5 consultas simples de una sola tabla:	10
--b). 2 actualizaciones y 2 borrados en cualquier tabla:	11
--c). 3 consultas con más de 1 tabla:	11
--d). 3 consultas usando funciones:.....	11
--e). 2 consultas usando GROUP BY:.....	12
--f). 2 consultas utilizando subconsultas:.....	12
--g). 2 consultas usando GROUP BY con HAVING:	12
--h). 3 actualizaciones usando subconsultas en WHERE y SET:	12
VISTAS:	13
FUNCIONES y CURSORES:	13
TRIGGERS:	13

PRIMARA PARTE

REQUISITOS:

En la reunión entre el agente digitalizador y la empresa van a obtener los requisitos que debe cumplir el software que se desarrollará.

Agente digitalizador: ¡Buenos días! Cuéntame a qué os dedicáis.

Gerente: Hola, somos una empresa y plataformas de entrega a domicilio. Queremos que realices el diseño de las bases de datos para una plataforma de entrega a domicilio. Para la eficiencia, después de una reunión de discusión, hemos establecido los siguientes requisitos:

- Nuestra plataforma se compone principalmente de tiendas, repartidores, almacenes, y proveedores. Para las tiendas, requerimos detalles como su NIF, nombre, categoría, dirección, número de cuenta bancaria y ventas. Además, considerando que las tiendas ofrecen productos, necesitamos información detallada de cada producto, incluyendo su número único, nombre, categoría, precio y nivel de inventario.

Para mejorar la experiencia del cliente, es crucial mantener un seguimiento de cada visita. Esto nos permitirá clasificar las tiendas y ofrecer recomendaciones personalizadas a los clientes.

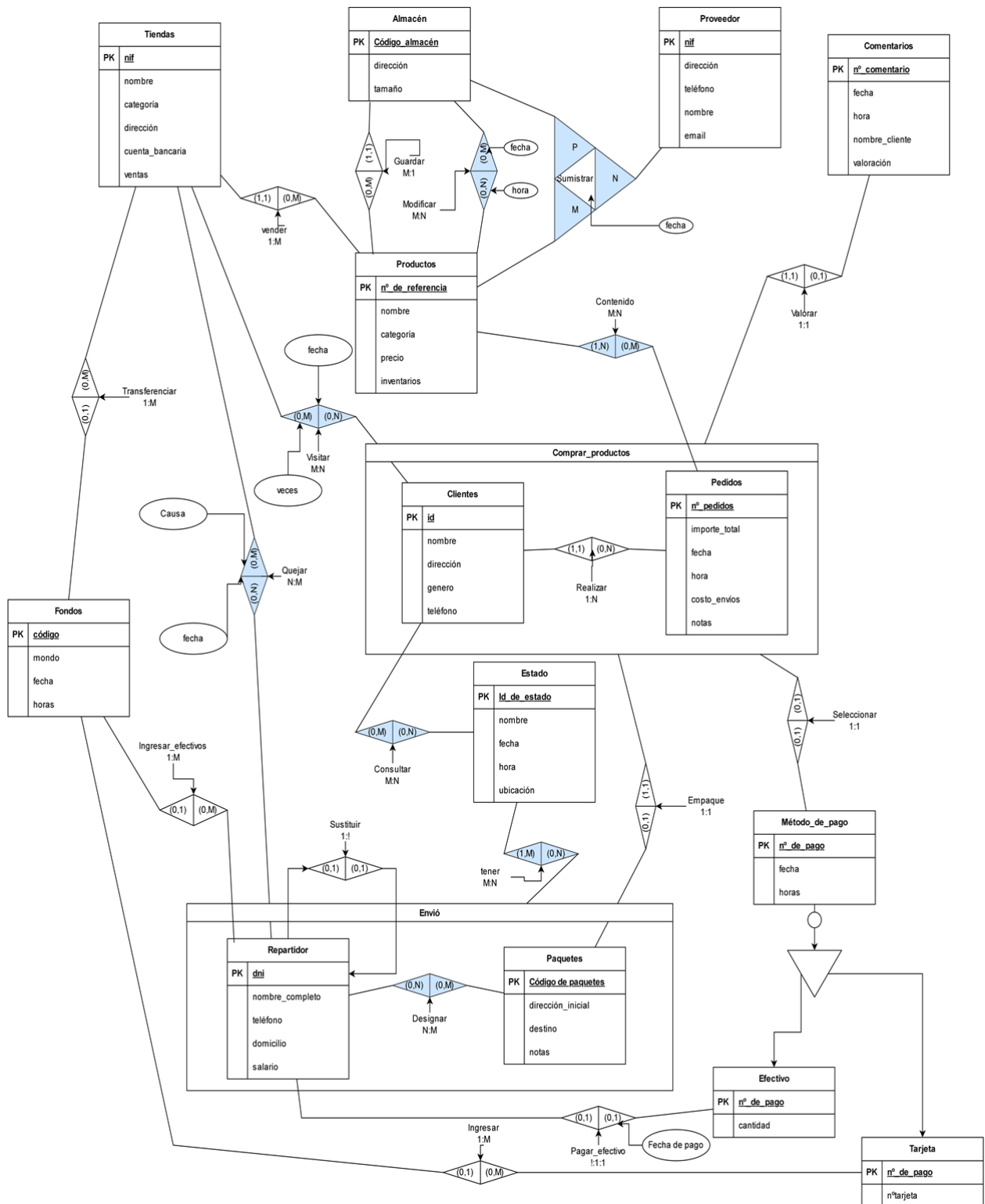
- Es importante mencionar que todos los productos se gestionarán en un almacén, y los empleados del almacén pueden realizar actualizaciones de información de los productos, registrando la fecha y hora de cada cambio. También necesitamos saber de qué almacén y proveedor proviene cada producto. En proveedor, requerimos su NIF, dirección, teléfono, nombre y correo electrónico.
- Ahora, hablemos de los pedidos. Cada pedido será creado por un cliente y seguramente incluirá al menos un producto. Con la creación de cada pedido se generará un número único, y se registrar su importe total, fecha, hora, costo de envío y notas asociadas.

- Después de que el cliente realice un pedido, podrá publicar un comentario (fecha, hora, nombre de cliente, puntuación). Al ejecutar el pedido, cliente puede elegir una forma de pago, seleccionando una opción. Si no elige ninguna, el pedido cancelará automáticamente.

De manera similar, después de que el cliente compre productos o realice un pedido, se creará como máximo un paquete. Esto se asignarán a uno o más repartidores, y cada paquete registrará la dirección de envío, destino y notas. Por si caso cada repartidor pueda tener un sustituidor o no.

- A continuación, existe dos formas de pago por son el efectivo y la tarjeta. Si es en efectivo, necesitamos saber la cantidad y la fecha de pago, ya que el efectivo se entregará al repartidor en primero, luego él lo depositará en la única estación de transferencia de fondos dentro de la fecha y hora establecidos por plataforma. Fondos registrará la fecha, hora y cantidad de cada transacción. En cuanto a la tarjeta, registraremos el número de tarjeta y el pago se transferirá directamente a la Fondo. Ambas formas de pago finalmente transferirán el monto a la cuenta de la tienda desde Fondo.
- Finalmente, las tiendas pueden presentar quejas contra los repartidores, pero deben indicar claramente el motivo, y también necesitamos guardar la fecha de la queja.

ESQUEMA DE ENTIDAD-RELACIÓN:



DISEÑO LÓGICO MODELO RELACIONAR:

Los que están escritas por mayúsculas son **entidades** y de minúsculas son **relaciones**.

FONDOS (código, cantidad, fecha_horas)

PK (código)

TIENDAS (nif, nombre, categoría, dirección, cuenta_bancaria, ventas, código_fondos)

PK (nif)

FK (código_fondos) → FONDOS

ALMACÉN (código_almacén, dirección, tamaño)

PK (código_almacén)

PRODUCTOS (nº_de_referencia, nombre, categoría, precio, inventarios, nif_tiendas, código_almacén)

PK (nº_de_referencia)

FK (nif_tiendas) → TIENDAS

FK (código_almacén) → ALMACÉN

VNN (tiendas)

VNN (almacén)

Modificar (código_almacén, nº_productos, fecha_hora)

PK (almacén, productos)

FK (código_almacén) → ALMACÉN

FK (nº_productos) → PRODUCTOS

PROVEEDOR (nif, dirección, teléfono, nombre, email)

PK (nif)

Suministrar (nif_proveedor, código_almacén, nº_productos, fecha)

PK (nif_proveedor, código_almacén, nº_productos)

FK (nif_proveedor) → PROVEEDOR

FK (código_almacén) → ALMACÉN

FK (nº_productos) → PRODUCTOS

CLIENTES (id, nombre, dirección, genero, teléfono)

PK (id)

Visitar (nif_tiendas, id_clientes, fechas, veces)

PK (nif_tiendas, id_clientes)

FK (nif_tiendas) → TIENDAS

FK (id_clientes) → CLIENTES

PEDIDOS (nºpedidos, importe_total, fecha_hora, costo_envíos, notas, id_cliente)

PK (nºpedidos)

FK (id_cliente) → CLIENTES

VNN (id_cliente)

Contenido (nºpedidos, nºproductos)

PK (nºpedidos, nºproductos)

FK (nºpedidos) → PEDIDOS

FK (nºproductos) → PRODUCTOS

COMENTARIOS (nºcomentarios, fecha_hora, valoración, nºpedidos, id_cliente)

PK (nºcomentarios)

UK (nºpedidos, id_cliente)

FK (nºpedidos) → PEDIDOS

FK (id_cliente) → CLIENTES

VNN (nºpedidos)

VNN (id_pedidos)

MÉTODO_DE_PAGO (nº_de_pago, fecha_hora, nºpedidos)

PK (nº_de_pago)

UK (nºpedidos)

FK (nºpedidos) → PEDIDOS

EFFECTIVO (nº_de_pago, cantidad, fecha, dni_repartidor)

PK (nº_de_pago)

FK (nº_de_pago) → MÉTODO_DE_PAGO

FK (dni_repartidor) → REPARTIDOR

TARJETA (nº_de_pago, nºtarjeta, fecha, código_fondo)

PK (nº_de_pago, nºtarjeta, fecha)

FK (nº_de_pago) → MÉTODO_DE_PAGO

FK (código_fondo) → FONDOS

Seleccionar (nºpedidos, id_cliente, nº_de_pago)

PK (nºpedidos)

UK (nº_de_pago, id_cliente)

FK (id_cliente) → CLIENTES

FK (nºpedidos) → PEDIDOS

FK (nº_de_pago) → MÉTODO_DE_PAGO

REPARTIDOR (dni, nombre_completo, teléfono, domicilio, salario, código_fondo, cantidad_ingresar)

PK (dni)

FK (código_fondo) → FONDOS

Sustituir (repartidor1, repartidor2)

PK (repartidor1)

UK (repartidor2)

FK (repartidor1) → REPARTIDOR

FK (repartidor2) → REPARTIDOR

Quejar (nif_tiendas, dni_repartidor, fecha, causa)

PK (nif_tiendas, dni_repartidor)

FK (nif_tiendas) → TIENDAS

FK (dni_repartidor) → REPARTIDOR

PAQUETES (código_de_paquetes, dirección_inicial, destino, notas, nºpedidos, id_cliente)

PK (código_de_paquetes)

UK (nºpedidos, id_cliente)

FK (nºpedidos) → PEDIDOS

FK (id_cliente) → CLIENTES

Designar (dni_repartidor, código_de_paquetes)

PK (dni_repartidor, código_de_paquetes)

FK (dni_repartidor) → REPARTIDOR

FK (código_de_paquetes) → PAQUETES

ESTADO (id_estado, nombre, fecha, hora, ubicación)

PK (id_estado)

Tener (id_estado, dni_repartidor, código_de_paquetes)

PK (id_estado, dni_repartidor, código_de_paquetes)

FK (id_estado) → ESTADO

FK (dni_repartidor, código_de_paquetes) → Designar

Consultar (id_cliente, id_estado)

PK (id_cliente, id_estado)

FK (id_cliente) → CLIENTES

FK (id_estado) → ESTADO

SEGUNDA PARTE

Para mejorar la estructura de las bases de datos, he realizado algunas modificaciones:

- Combine fechas y horas de algunas tablas en un solo atributo.
- Para adaptarse a la realidad, se asegura de que ciertos atributos en algunas tablas es valor no nulo.
- Con el fin de garantizar la creación de la base de datos sin problemas, se eliminarán o sustituirán algunas letras especiales y acentos, como ñ, á, é, í..., para prevenir posibles inconvenientes.

COMO RESULTADO, LAS TABLAS PRESENTAN DIFERENTES CON RESPECTO AL MODELO LÓGICO Y DE RELACIONES ORIGINAL...

Para obtener detalles adicionales, consulte el archivo 'DDL.sql' correspondiente.

LAS CONSULTAS REALIZADOS:

--A). 5 CONSULTAS SIMPLES DE UNA SOLA TABLA:

- 1. Obtener todos los productos de la categoría 'Electrónicos'.
- 2. Obtener la información de todos almacenes cuyo su tamaño es mayor que 600 metros cuadrados.
- 3. Obtener la lista de todos los pedidos realizados.
- 4. Obtener los comentarios y valoraciones de los pedidos.
- 5. Obtener el nombre y email de todos los proveedores.

--B). 2 ACTUALIZACIONES Y 2 BORRADOS EN CUALQUIER TABLA:

- 1. Actualizar la dirección del cliente con ID 'C001'.
- 2. Aumentar el precio de todos los productos en un 10%.
- 3. Borrar el pedido con número 'PED003'.
- 4. Borrar el comentario con número 'COM002'.

--C). 3 CONSULTAS CON MÁS DE 1 TABLA:

- 1. Obtener información de pedidos y productos relacionados.
- 2. Mostrar las tiendas y los clientes que las visitaron.
- 3. Obtener detalles de los paquetes y los estados actuales.

--D). 3 CONSULTAS USANDO FUNCIONES:

- 1. Calcular el total de ventas de todas las tiendas.
- 2. Calcular el promedio de valoraciones de los comentarios.
- 3. Obtener la diferencia de horas entre la fecha de inicio y fin de sustitución de repartidores.

--E). 2 CONSULTAS USANDO GROUP BY:

- 1. Calcular la cantidad total de productos por categoría en la tabla PRODUCTOS.
- 2. Contar la cantidad de visitas por tienda en la tabla Visitar.

--F). 2 CONSULTAS UTILIZANDO SUBCONSULTAS:

- 1. Obtener los pedidos realizados por clientes nombres mayor de 3 dígitos.
- 2. Mostrar los repartidores con salario superior al promedio.

--G). 2 CONSULTAS USANDO GROUP BY CON HAVING:

- 1. Filtrar las tiendas con menos de 100 visitas en la tabla Visitar.
- 2. Obtener las categorías de productos con inventario total superior a 100.

--H). 3 ACTUALIZACIONES USANDO SUBCONSULTAS EN WHERE Y SET:

- 1. Actualizar la cuenta bancaria de las tiendas con ventas superiores a 700.
- 2. Actualizar el precio de los productos con un 20% de descuento.
- 3. Actualizar la cantidad de fondos después de un pedido (restando el importe total del pedido).

TERCERA PARTE

VISTAS:

Diseñar dos vistas útiles para el proyecto. Debes explicar el diseño y exponer una justificación de creación.

VISTA 1. VISTA PEDIDOS DETALLE:

- **Diseño:** Esta vista muestra detalles completos de los pedidos, incluyendo la información del cliente, productos pedidos y el repartidor asignado.
- **Justificación:** Esta vista permite a los usuarios ver fácilmente todos los detalles de un pedido sin necesidad de realizar múltiples consultas de unión entre tablas. Esto es muy útil para el servicio al cliente y la gestión de pedidos.

VISTA 2. VISTA PRODUCTOS INVENTARIO:

- **Diseño:** Esta vista muestra todos los productos y su estado actual de inventario.
- **Justificación:** Esto ayuda en la gestión de inventarios, permitiendo a los administradores de inventario ver rápidamente el estado del inventario de los productos para reabastecer o ajustar el stock oportunamente.

FUNCIONES Y CURSORES:

Realiza cuatro funciones útiles para el proyecto. En una de ellas mínimo debes utilizar un cursor.

FUNCIONES 1. OBTENER REPARTIDOR DISPONIBLE:

- **Funcionalidad:** Obtener un repartidor disponible en el momento actual.
- **Razón:** Automatiza la asignación de repartidores para mejorar la eficiencia.

FUNCIONES 2. CALCULAR DESCUENTO:

- **Funcionalidad:** Calcular el monto de descuento para un pedido según un porcentaje dado.
- **Razón:** Automatiza el cálculo del monto de descuento, facilitando las actividades promocionales.

FUNCIONES 3. ACTUALIZAR ESTADO PEDIDO:

- Funcionalidad: Actualizar el estado de un pedido.
- Razón: Unifica la actualización del estado del pedido a través de una función para asegurar la consistencia de la lógica del negocio.

FUNCIONES 4. LISTAR PRODUCTOS CATEGORÍA:

- Funcionalidad: Usar un cursor para listar todos los productos de una categoría específica.
- Razón: Usar un cursor para procesar filas individualmente puede mejorar la eficiencia al manejar grandes conjuntos de datos.

TRIGGERS:

Realiza cuatro 'Triggers' en el proyecto, deben haber 'triggers' de los dos tipos siguientes:

- 1) **'Trigger' global que afecte a la tabla que quieras y que haga algún tipo de validación global que no permita alguna operación de insertar, de borrado o de modificación.**

TRIGGER 1. NO BORRAR CLIENTE CON PEDIDOS:

- Tabla Afectada: Clientes
- Operaciones: DELETE
- Funcionalidad: Impide eliminar un cliente que tiene pedidos asociados.
- Razón: Mantiene la integridad de los datos al evitar la eliminación de registros importantes para el negocio. Si un cliente tiene pedidos asociados, su eliminación podría llevar a inconsistencias y pérdida de información crucial.

- 2) **'Trigger' que vaya por fila, y que haga una automatización que consideres en tu aplicación. Por ejemplo, cuando hagas una operación sobre una tabla debe automáticamente hacer algo sobre otra tabla.**

TRIGGER 2. ACTUALIZAR STOCK:

- Tabla Afectada: Pedidos
- Operaciones: INSERT
- Funcionalidad: Actualiza automáticamente el inventario al insertar un pedido.
- Razón: Automatiza la gestión del inventario, reduciendo errores en la actualización manual del inventario. Esto garantiza que el inventario refleje con precisión la cantidad disponible después de cada nueva orden.

TRIGGER 3. RECALCULAR TOTAL PEDIDO

- Tabla Afectada: Pedidos
- Operaciones: UPDATE
- Funcionalidad: Recalcula el monto total del pedido al actualizar un pedido.
- Razón: Asegura que el monto total del pedido sea siempre exacto, reflejando cualquier cambio en las cantidades o precios de los productos en el pedido. Esto es crucial para mantener la precisión en la facturación y el análisis financiero.

TRIGGER 4: VALIDAR FORMATO TELÉFONO:

- Tabla Afectada: Clientes
- Operaciones: INSERT, UPDATE
- Funcionalidad: Valida el formato del número de teléfono al insertar o actualizar un cliente.
- Razón: Garantiza la correcta formatearían de los números de teléfono, manteniendo la consistencia de los datos. Esto es importante para la comunicación eficiente con los clientes y para evitar problemas de formato que podrían dificultar la gestión de contactos.