

PROYECTO **NEW ORDER**

Comisión 2 - Grupo 03

INTEGRANTES:

- 1.BECK, ANA PAULA
- 2.RAMOS, MILAGROS
- 3.SOTELO, FABIO
- 4.TITONEL, LUCAS

INTRODUCCIÓN AL PROYECTO

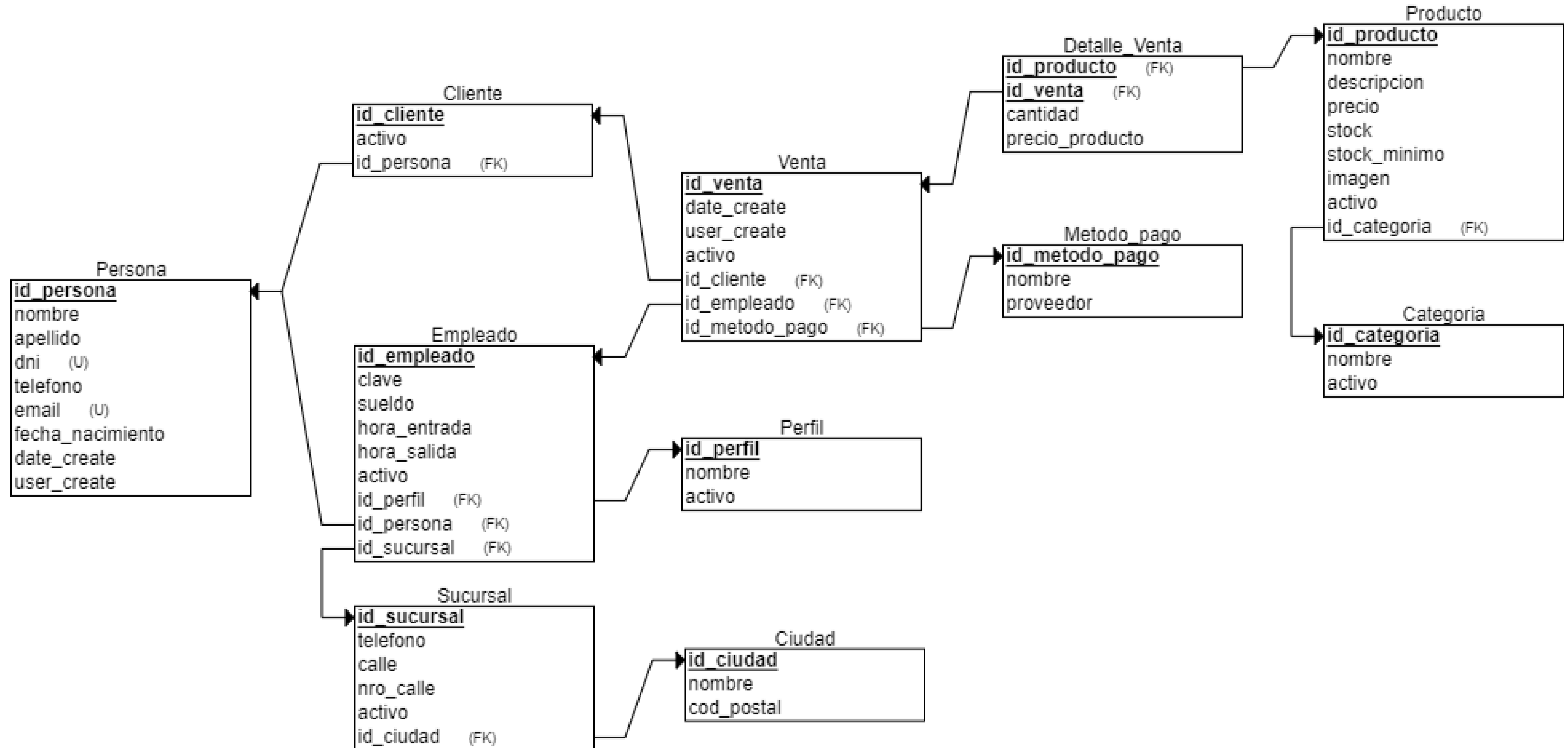
El sistema de gestión de ventas está diseñado para tiendas físicas. Está orientado a registrar y administrar de forma manual las ventas, clientes, productos y otros datos correspondientes a un comercio.

El sistema cuenta con módulos específicos asignados según el perfil del personal de la empresa, destacando los roles como:

- gerente-->acceso a la gestión del personal, clientes, productos e informes.
- vendedor-->administración de clientes, la gestión de productos y el procesamiento de ventas.
- administrador-->respaldo y la seguridad de los datos, gestionando el backup.



MODELO RELACIONAL



TEMAS

01

Manejo de permisos a nivel de usuarios de base de datos.

02

Procedimientos y funciones almacenadas.

03

Optimización de consultas a través de índices.

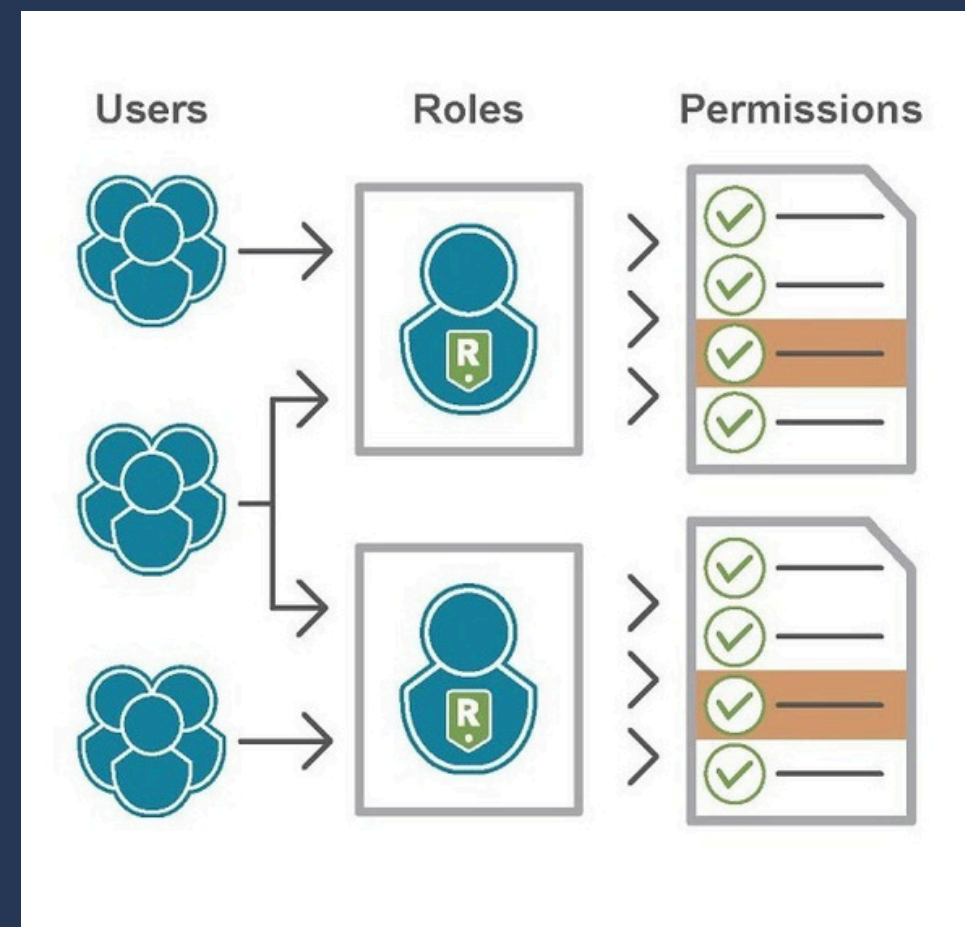
04

Manejo de tipos de datos JSON.

MANEJO DE PERMISOS A NIVEL DE USUARIOS DE BASE DE DATOS

Permisos

es una autorización específica que se concede a un usuario o grupo de usuarios para realizar una acción particular en un objeto de la base de datos (como tablas, vistas o procedimientos almacenados).



Roles

conjunto de permisos que se asigna como una unidad a uno o más usuarios. Los roles permiten agrupar permisos comunes bajo un nombre, simplificando la administración de accesos.

MANEJO DE PERMISOS A NIVEL DE USUARIOS DE BASE DE DATOS

1

CREACION DE USUARIOS CON PERMISOS DIFERENCIADOS

- 1.1 **Usuario con Permiso de Administrador** (UsuarioAdmin): Este usuario fue asignado al rol db_owner, que le permite acceso completo a la base de datos.
- 1.2 **Usuario con Permiso de Solo Lectura** (UsuarioLectura): Este usuario solo tiene permisos de lectura sobre una tabla en particular.

2

ASIGNACION DE PERMISOS DE EJECUCION DE PROCEDIMIENTOS ALMACENADOS

- 2.1 **A UsuarioLectura** se le otorgó permiso de ejecución sobre el procedimiento creado, permitiéndole insertar datos en Producto indirectamente, sin acceso directo a la tabla

MANEJO DE PERMISOS A NIVEL DE USUARIOS DE BASE DE DATOS

3 CREACION DE ROLES PARA SIMPLIFICAR LA GESTION DE PERMISOS

3.1 **Se creó el rol RolSoloLectura**, que otorga permisos de lectura en la tabla Producto.

Se crearon dos usuarios adicionales:

- **UsuarioRol1, asignado al rol RolSoloLectura**, tiene permisos para realizar consultas Producto pero no puede modificar ni insertar datos.
- **UsuarioRol2 no tiene ningún rol asignado**, por lo que no tiene permisos de acceso a Producto.

PROCEDIMIENTOS Y FUNCIONES ALMACENADAS

Procedimientos Almacenados

Conjunto de instrucciones que se almacenan en la base de datos y que pueden ser ejecutados como una unidad. Permiten realizar operaciones complejas, (consultas, manipulaciones de datos y operaciones condicionales)

Funciones Almacenadas

Conjunto de instrucciones que se guarda en la base de datos y que siempre devuelve un valor como resultado.
Tienen limitaciones, como no poder modificar datos directamente en la base de datos.

PROCEDIMIENTOS Y FUNCIONES ALMACENADAS

1

CREACION DE PROCEDIMIENTOS
ALMACENADOS (operaciones CRUD)

1.1 **Procedimiento para la inserción de un producto (insertarProducto):**

Permite agregar nuevos productos especificando parámetros como nombre, descripción, precio, stock, entre otros.

1.2 **Procedimiento para la modificación de un producto (modificarProducto):**

Permite actualizar los datos de un producto existente.

1.3 **Procedimiento para la eliminación de un producto (borrarProducto):**

Permite eliminar un producto de la base de datos.

PROCEDIMIENTOS Y FUNCIONES ALMACENADAS

2 INSERCIÓN DE DATOS MEDIANTE PROCEDIMIENTOS

2.1 Se realizaron inserciones de productos tanto con sentencias **INSERT directas** como mediante la ejecución de **procedimientos almacenados**. Se incluyeron 25 registros con cada método para comparar su eficiencia.

3 CREACIÓN DE FUNCIONES ALMACENADAS

3.1 **calcularDescuento:** Calcula un descuento sobre el precio de un producto dado un porcentaje.

3.2 **esStockBajo:** Verifica si el stock de un producto está por debajo del mínimo.

3.3 **esProductoActivo:** Indica si un producto está activo o inactivo.

3.4 **obtenerCategoriaProducto:** Devuelve el nombre de la categoría a la que pertenece un producto.

PROCEDIMIENTOS Y FUNCIONES ALMACENADAS

4 COMPARACION DE EFICIENCIA Y CONCLUSIONES

4.1 **Se comparó la eficiencia de las operaciones directas sobre las tablas versus el uso de procedimientos y funciones.** Los procedimientos proporcionan mayor seguridad y encapsulamiento de lógica, facilitando operaciones complejas y reutilización del código. Se observó un ligero aumento en el tiempo de ejecución para inserciones mediante procedimientos debido al procesamiento adicional, pero este fue compensado por la mejora en la consistencia y mantenimiento.

4.2 **El uso de procedimientos y funciones almacenadas debe planearse con atención,** buscando maximizar sus ventajas cuando la lógica en el servidor sea crítica, pero sin comprometer la flexibilidad, el mantenimiento o la capacidad de evolución del sistema. La decisión debe alinearse con las necesidades específicas del proyecto, el entorno de despliegue y los objetivos de rendimiento y seguridad, asegurando un equilibrio adecuado entre simplicidad, control y eficiencia.

OPTIMIZACION DE CONSULTAS A TRAVES DE INDICES

Agrupados

- Ordena físicamente los datos en la tabla según el campo o conjunto de campos.
- Cada tabla solo puede tener un índice agrupado.
- Contienen los datos reales
- Suele usarse en consultas frecuentemente sobre un rango o claves primarias.

No Agrupados

- Los datos de la tabla no se organizan físicamente sino que están dispersos.
- Una tabla puede tener múltiples índices no agrupados.
- Contiene el puntero que especifica donde se almacenan los datos.
- Suele usarse para consultas específicas o filtros que requieren acceso rápido.

OPTIMIZACIÓN DE CONSULTAS A TRAVÉS DE INDICES

1

REALIZAR UNA CARGA MASIVA DE
1 MILLON DE DATOS

1.1

Se utilizan tablas, variables y cálculos temporales para el tratamiento de las fechas y mediante un bucle se crean este lote de registros.

2

CONSULTA DE LOS DATOS A PARTIR
DE BUSQUEDA POR PERIODO

2.1

Habilitar las sentencias para medir el tiempo de respuesta y las operaciones de E/S de la consulta a realizar

2.2

Realizar una consulta sobre la tabla "Ventas" donde se rescatan las ventas por periodo de fecha de 1 mes

2.3

Revisar los resultados registros devueltos, el tiempo de rta y recursos consumidos. Deshabilitar las sentencias



OPTIMIZACIÓN DE CONSULTAS A TRAVÉS DE INDICES

3 CREAR INDICE AGRUPADO POR MEDIO DEL CAMPO FECHA REALIZAR CONSULTAS Y COMPARAR TIEMPO DE EJECUCION

- 3.1 Eliminar el índice agrupado actual (clave primaria en campo id_venta)
- 3.2 Crear un nuevo índice agrupado sobre la columna "fecha_venta"
- 3.3 Crear un nuevo índice no agrupado sobre la columna "id_venta" para mantener la integridad referencial de clave primaria.
- 3.4 Repetir la consulta realizada en el item 2 y comparar los resultados obtenidos en el tiempo de respuesta

4 ELIMINAR EL INDICE CREADO

Se ocupa la sentencia "DROP INDEX" indicando el nombre del índice y accediendo a la tabla que queremos borrarlo.



OPTIMIZACIÓN DE CONSULTAS A TRAVÉS DE INDICES

5 DEFINIR INDICE NO AGRUPADO SOBRE LA COLUMNA FECHA Y QUE ADEMÁS INCLUYA LAS COLUMNAS SELECCIONADAS.
REPETIR CONSULTA

5.1 Crear un índice no agrupado que incluya todas las columnas de la tabla "Ventas2"

5.2 Repetir la consulta realizada en el ítem 2 y comparar los resultados obtenidos en el tiempo de respuesta

5.3 Comparar de tiempos de rta y plan de ejecución entre tablas (venta1, sin índice) y (ventas2, con índice no agrupado)

6 CONCLUSIONES

Las consultas sin índices necesitan hacer un escaneo completo de la tabla, lo que conlleva un alto costo de rendimiento.

Las consultas tanto con índices agrupados como no agrupados, reducen el número de páginas leídas y mejora el rendimiento de búsqueda, permitiendo consultas rápidas y eficientes.



MANEJO DE TIPOS DE DATOS CON JSON

¿Qué es JSON?

JSON es un formato ligero y de texto para el intercambio de datos. Se estructura en pares clave-valor y es ampliamente utilizado para transmitir datos entre aplicaciones web y servidores.

Implementacion en BD relacionales

En BD relacionales, se usa para almacenar y manejar datos semiestructurados dentro de campos específicos, permitiendo flexibilidad en la representación de datos que no encajan bien en el modelo de tablas tradicionales.

MANEJO DE TIPOS DATOS CON JSON

1

CREAR UNA NUEVA TABLA CON UNA COLUMNA JSON

2

AGREGAR UN CONJUNTO DE DATOS NO ESTRUCTURADOS EN FORMATO JSON, Y REALIZAR OPERACIONES DE ACTUALIZACIÓN, AGREGACIÓN Y BORRADO DE DATOS.

3

REALIZAR OPERACIONES DE CONSULTAS.

4

APROXIMACIONES A LA OPTIMIZACIÓN DE CONSULTAS PARA ESTAS ESTRUCTURAS





**MUCHAS
GRACIAS**

