**Optimización de la Base de Datos de Ventas – Tienda Virtual**

**INDICES**

***Definición:***

* Los índices actúan como accesos rapidos
* Evita que la base de datos tenga que buscar fila por fila
* Acelera consultas

1. **Índice en la tabla DetalleVenta**

**CREATE INDEX idx\_DetalleVenta\_PedidoID ON DetalleVenta(PedidoID);**

**Razon:** La tabla DetalleVenta puede tener muchos productos para un solo pedido

**Ejemplo:**

Yo cliente pido 2 laptops y 4 audífonos = 1 solo pedido y muchos productos

**SELECT \* FROM DetalleVenta WHERE PedidoID = 1;**

* Sin índice SQL revisa toda la tabla
* Con índice va directo a los registros de PedidoID = 1

**¿Porque es importante?**

* La tienda usara todo el tiempo esa CONSULTA para ver que producto hay en cada pedido
* Un índice en PedidoID hace que la búsqueda sea instantánea, en lugar de recorrer miles de filas

1. **Índice en la tabla Pagos**

**CREATE INDEX idx\_Pagos\_PedidoID ON Pagos(PedidoID);**

**Razon:** La tabla pagos almacena los pagos de cada pedido

**Ejemplo:**

Yo dueño quiero saber cuanto pago tal cliente por tal pedido

**SELECT \* FROM Pagos WHERE PedidoID = 10;**

* Sin índice revisa todos lo pagos uno por uno
* Con índice va directo al pedidoID

**¿Porque es importante?**

* Cada pedido puede tener múltiples pagos
* Ejemplo: un cliente paga una parte con tarjeta y la otra parte en efectivo
* El índice hace que la consulta sea mucho más rápida cuando el sistema quiere ver pagos de un pedido

**Beneficios de los Índices:**

* Aceleran las consultas, las búsquedas y filtrados son más rápidos.
* Mejoran el rendimiento, estos reducen el tiempo de ejecución de consultas frecuentes.
* Optimizan el uso de recursos, disminuyen la carga en el servidor de bases de datos.

**TRANSACCIONES**

***Definición:***

Las transacciones son secuencias de operaciones que se ejecutan como una sola unidad de trabajo. Si una operación falla, todas las operaciones dentro de la transacción se deshacen (ROLLBACK). Esto garantiza la integridad de los datos.

**STORED PROCEDURE**

***Definición:***

Un Stored Procedure (Procedimiento Almacenado) es un conjunto de sentencias SQL que se almacenan en la base de datos y se ejecutan como una unidad. Son útiles para encapsular lógica de negocio y reutilizar código.

* **Stored Procedure para consultar ventas por Cliente:**

**CREATE PROCEDURE ConsultarVentasPorCliente**

**@ClienteID INT**

**AS**

**BEGIN**

**SELECT**

**P.PedidoID,**

**P.FechaPedido,**

**P.Estado,**

**P.Total,**

**DV.ProductoID,**

**DV.Cantidad,**

**DV.PrecioUnitario,**

**PR.NombreProducto**

**FROM Pedidos P**

**INNER JOIN DetalleVenta DV ON P.PedidoID = DV.PedidoID**

**INNER JOIN Productos PR ON DV.ProductoID = PR.ProductoID**

**WHERE P.ClienteID = @ClienteID;**

**END;**

Para ejecutar el stored Procedure y obtener las ventas de un cliente específico, usamos el comando:

**EXEC ConsultarVentasPorCliente @ClienteID = 1;**

Esto devuelve todos los pedidos realizados por el cliente con ClienteID, junto con los detalles de los productos vendidos en cada pedido.