|  |
| --- |
| SENA CEET |
| Entregable #2 |
| En el proyecto se evidencia el uso de la base de datos a través de sentencias DML (SQL), revisando existencia de datos de prueba insertados, Joins, consultas y subconsultas. |

Miguen Ángel Gómez Sepúlveda

Jordán Michell Castañeda Lozano

Henry Alejandro Castro Torres

Jason Andres Espinosa Beltrán

Joseph Donovan Socha Bonilla

Presentación sentencias DML (Data Manipulation Languaje).

Una sentencia DML son un conjunto de comandos SQL para gestionar y manipular los datos dentro de una base de datos, para gestionar datos de manera más coherente vamos a utilizar KPIS (Key Perfomance Indicators).

Una KPI (Indicadores claves de rendimiento) son indicadores que se utilizan para medir el éxito y rendimiento de algunos procesos o actividades, los KPI los extraeremos de la construcción de la base de datos mediante de las sentencias DDL (Data Definition Language).

De las sentencias DDL pudimos extraer 13 KPI relevantes, las cuales son las siguientes:

1. La primera KPI es la de Total de productos en el inventario.

SELECT COUNT(\*) AS TotalProductos

FROM Producto;

Con esto podremos consultar cuantos productos hay en el inventario.

1. La segunda KPI es del promedio del stock por producto.

SELECT AVG(CAST(Stock AS INT)) AS InventarioPromedio

FROM Inventario;

Con esto podremos consultar cual es el stock promedio entre todos los productos del inventario.

1. La tercera KPI es el total de pedidos generados.

SELECT COUNT(DISTINCT Pedido\_NumPedido) AS TotalPedidos

FROM Detalle\_Pedido;

Con esto podremos consultar el total de pedidos que se han realizado

1. La cuarta KPI la podremos utilizar para consultar cuales productos tienen bajo stock menor o igual a 2 y cual es su id.

SELECT IdInventario, Stock FROM Inventario WHERE Stock <= 2;

Con esta consulta podemos analizar de mejor manera los niveles bajos de stock, junto con su id para poder reconocerlos de mejor manera.

1. La quinta KPI será para saber cuántos pedidos están asociados a cada vendedor con su respectivo nombre, en orden de cual tiene mas agendados y cuales no tienen ningún pedido asignado.

SELECT V.DniEmpleado, E.NombreEmpleado,

COUNT(P.NumPedido) AS TotalPedidos

FROM Vendedor V

LEFT JOIN Vendedor\_Pedido VP ON V.DniEmpleado = VP.Empleado\_DniEmpleado

LEFT JOIN Pedido P ON VP.Pedido\_NumPedido = P.NumPedido

LEFT JOIN Empleado E ON V.DniEmpleado = E.DniEmpleado

GROUP BY V.DniEmpleado, E.NombreEmpleado

ORDER BY TotalPedidos DESC;

De esta manera podemos observar de manera mas optima cuales empleados tienen mas pedidos agendados y por el contrario cuales no tienen pedidos.

1. Para la sexta KPI organizaremos la cantidad de pedidos por días.

SELECT FechaCreacion\_Factura, SUM(PrecioDetalleFactura \* CantidadDetalleFactura) AS TotalVentas

FROM Detalle\_Factura

JOIN Factura ON Detalle\_Factura.Factura\_IdFactura = Factura.IdFactura

GROUP BY FechaCreacion\_Factura

ORDER BY FechaCreacion\_Factura;

Con esto podremos comprobar cuales días son en los que más se hacen pedidos.

1. La séptima KPI que vamos a utilizar es la de listar que cantidad de veces se usaron los diferentes métodos de pagos.

SELECT Metodo\_Pago.NombreMetodoPago, COUNT(DISTINCT Cliente.DniCliente) AS TotalClientes

FROM Metodo\_Pago

JOIN Factura ON Metodo\_Pago.Factura\_IdFactura = Factura.IdFactura

JOIN Cliente ON Factura.Cliente\_DniCliente = Cliente.DniCliente

GROUP BY Metodo\_Pago.NombreMetodoPago;

Con esto podemos observar con mayor claridad cuáles son los métodos de pago que más se usan.

1. La octava KPI la usaremos para listar los 15 productos con mayor stock que 5 en el inventario.

SELECT TOP 15 IdInventario, Stock

FROM Inventario

WHERE Stock >5

ORDER BY Stock DESC ;

Con esto se tendrá una mayor claridad a la hora de saber que productos ya tienen un stock optimo para no hacer sobreproducción de productos.

1. Para la novena KPI obtendremos el número total de productos que están agotados.

SELECT COUNT(\*) AS ProductosAgotados FROM Inventario WHERE Stock = 0;

Esto nos permitirá saber que está agotado y que debe fabricarse.

1. Para la decima KPI obtendremos el total de ventas o pedidos por producto.

SELECT Producto\_IdProducto, SUM(PrecioDetalleFactura \* CAST(CantidadDetalleFactura AS DECIMAL(10,3))) AS VentasTotales

FROM Detalle\_Factura

GROUP BY Producto\_IdProducto;

Este indicador nos ayudará a identificar cual es el producto más vendido y por ende cual se debe fabricar más.

1. Para la onceava KPI listaremos la cantidad de productos por cada categoría.

SELECT Categoria\_IdCategoria, COUNT(\*) AS TotalProductos

FROM Categoria\_Producto

GROUP BY Categoria\_IdCategoria;

Esto nos facilitará conocer de manera más rápida y precisa cuantos productos debemos tener por categoría.

1. La doceava KPI será para consultar si un cliente ha realizado más de un pedido.

SELECT Cliente\_DniCliente, COUNT(\*) AS TotalPedidos

FROM Pedido

GROUP BY Cliente\_DniCliente

HAVING COUNT(\*) >1;

Esto nos ayudará a llevar una trazabilidad de los pedidos de los clientes para fidelizar aquellos que llevan más de cierta cantidad de compras.

1. La treceava KPI nos indicara una cierta cantidad de los productos más vendidos

SELECT TOP 3 Producto\_IdProducto, SUM(CAST(CantidadDetalleFactura AS INT)) AS TotalVendidos

FROM Detalle\_Factura

GROUP BY Producto\_IdProducto

ORDER BY TotalVendidos DESC;

Esto facilitará que los clientes puedan identificar los productos más vendidos y poder facilitar el proceso de compra.