

BLOQUE 1

EJERCICIO 1

Crea un procedimiento almacenado `ProductosPorPrecio` que reciba un precio mínimo y devuelva todos los productos cuyo `PrecioUnitario` sea mayor o igual a ese valor, ordenados de mayor a menor.

EXEC ProductosPorPrecio 40

100 %

Results Messages

| | ProductoID | Nombre | PrecioUnitario |
|---|------------|------------------------|----------------|
| 1 | 10 | Monitor 24 pulgadas | 150.00 |
| 2 | 31 | Disco duro externo 1TB | 70.00 |
| 3 | 8 | Teclado mecánico | 60.00 |
| 4 | 14 | Chaqueta impermeable | 45.00 |

EJERCICIO 2

- 1) Define la función escalar `dbo.PrecioConIVA` que agregue 13 % de IVA a un monto.
- 2) Implementa el procedimiento `DetalleFacturaConIVA` que reciba `@FacturaID`, recorra sus líneas con un cursor, imprima el subtotal con IVA usando la función y, al final, muestre el total con IVA.

EXEC DetalleFacturaConIVA 13

100 %

Messages

```
Detalle 81 Subtotal c/IVA: $27.12
Detalle 82 Subtotal c/IVA: $13.56
Detalle 83 Subtotal c/IVA: $126.56
-----
TOTAL + IVA: $167.24
```

EJERCICIO 3

Desarrolla el procedimiento `FacturasMes` que reciba `@MM` y `@YYYY`, recorra con cursor todas las facturas de ese mes y año, imprima su monto y acumule el total mensual.

```
EXEC FacturasMesCursor 01, 2025
```

100 %

Messages

| | |
|-------------|------------|
| Factura 377 | \$732.70 |
| Factura 382 | \$284.20 |
| Factura 384 | \$246.75 |
| Factura 388 | \$120.80 |
| Factura 397 | \$500.30 |
| Factura 398 | \$392.20 |
| ----- | |
| TOTAL MES: | \$22073.20 |

Completion time: 2025-06-23T23:21:27.7817315-06:00

BLOQUE 2

EJERCICIO 1

Crea el procedimiento `InsertarProducto` que reciba Nombre, Precio y CategoricalID, genere el siguiente `ProductoID` secuencial y registre el producto con un stock inicial de 100.

```
EXEC InsertarProducto 'Producto prueba', 10.0, 1
```

100 %

Messages

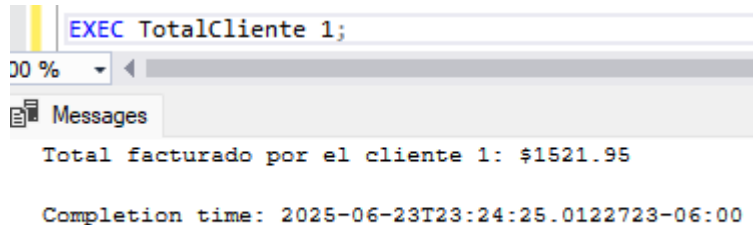
(1 row affected)

(1 row affected)

Completion time: 2025-06-23T23:22:31.6831289-06:00

EJERCICIO 2

- 1) Crea la función tabla `dbo.FacturasDeCliente(@ClienteID)` que devuelva `FacturalID`, `FechaFactura` y `Monto`.
- 2) Diseña el procedimiento `TotalCliente` que use la función para calcular y mostrar el total facturado a ese cliente.



```
EXEC TotalCliente 1;
```

100 %

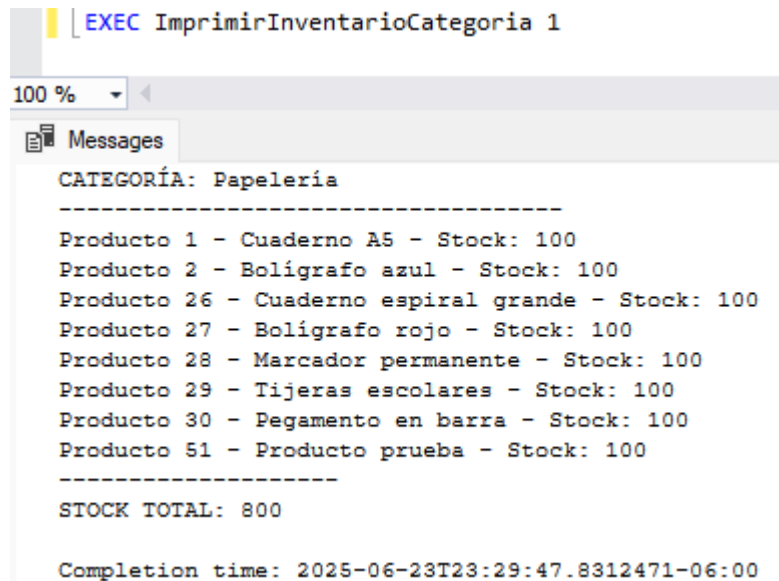
Messages

Total facturado por el cliente 1: \$1521.95

Completion time: 2025-06-23T23:24:25.0122723-06:00

EJERCICIO 3

Implementa el procedimiento `ImprimirInventarioCategoria` que reciba `CategorialD`, recorra con cursor todos los productos de esa categoría, imprima nombre y stock, y al final muestre el stock total de la categoría.



```
EXEC ImprimirInventarioCategoria 1
```

100 %

Messages

CATEGORÍA: Papelería

Producto 1 - Cuaderno A5 - Stock: 100

Producto 2 - Bolígrafo azul - Stock: 100

Producto 26 - Cuaderno espiral grande - Stock: 100

Producto 27 - Bolígrafo rojo - Stock: 100

Producto 28 - Marcador permanente - Stock: 100

Producto 29 - Tijeras escolares - Stock: 100

Producto 30 - Pegamento en barra - Stock: 100

Producto 51 - Producto prueba - Stock: 100

STOCK TOTAL: 800

Completion time: 2025-06-23T23:29:47.8312471-06:00

BLOQUE 3 (Avanzados)

EJERCICIO 1

Define un procedimiento ResumenVentasPorCategoria que recorra todas las categorías de productos y, para cada una, calcule e imprima:

- El número total de productos vendidos (sumando las cantidades en DetalleFactura)
- El monto total recaudado
- El promedio de venta por producto

```
EXEC ResumenVentasPorCategoria
```

```
100 %
```

```
Messages
```

```
Categoría: Papelería
  Total unidades vendidas: 938
  Total recaudado: $2410.45
  Promedio por unidad: $2.57
-----
Categoría: Electrónica
  Total unidades vendidas: 1533
  Total recaudado: $64014.50
  Promedio por unidad: $41.76
-----
Categoría: Ropa
  Total unidades vendidas: 1548
  Total recaudado: $28610.50
  Promedio por unidad: $18.48
-----
Categoría: Alimentos
  Total unidades vendidas: 1557
  Total recaudado: $5028.00
  Promedio por unidad: $3.23
-----
Categoría: Higiene
  Total unidades vendidas: 1575
  Total recaudado: $5972.60
  Promedio por unidad: $3.79
-----
```

EJERCICIO 2

- 1- Diseña una función con valor de tabla llamada `dbo.TotalFacturadoConIVAClientes` que calcule, para cada cliente, el monto total facturado incluyendo el 13% de IVA.
Para ello, reutiliza una función escalar existente llamada `dbo.PrecioConIVA(@Precio)`.
- 2- Luego, implementa un procedimiento almacenado llamado `RankingClientesPorIngreso` que utilice dicha función para:
 - Mostrar el Top 5 de clientes con mayor facturación total (con IVA).
 - Imprimir únicamente el nombre, apellido y el total facturado con IVA, ordenado de mayor a menor.

EXEC RankingClientesPorIngreso

100 %

Results Messages

| | Nombre | Apellido | TotalConIVA |
|---|------------|------------|-------------|
| 1 | Gloria | Herrera A. | 2673.47 |
| 2 | Julieta | Jurado | 2495.50 |
| 3 | Natalia A. | Navarro A. | 2495.22 |
| 4 | Queralt | Rosales | 2197.51 |
| 5 | Jacobo | Klein | 2142.60 |