

Documentación Consultas a la Base de Datos

En el proyecto se evidencia el uso de la base de datos a través de sentencias DDL (SQL), revisando existencias de datos prueba insertados, Joins, consultas y subconsultas o MongoDB CRUD Operations y aggregations.

1. Consulta de Actividad y Días Activos de Usuarios:

```
SELECT
  u.nombre_usuario,
  COUNT(ra.id_registro) AS total_actividades,
  CASE
    WHEN COUNT(ra.id_registro) = 0 THEN 0
    ELSE DATEDIFF(DAY, MIN(ra.fecha_actividad), MAX(ra.fecha_actividad))
  END AS dias_activos
FROM
  usuarios u
LEFT JOIN
  registros_actividades ra ON u.id_usuario = ra.id_usuario
GROUP BY
  u.nombre_usuario;
```

nombre_usuario	total_actividades	dias_activos
admin	4	0
almacenista1	2	0
analista1	0	0
asistente1	1	0
auditor1	1	0

2. Ventas Totales por Mes

```
SELECT
  YEAR(f.fecha_emision_factura) AS año,
  MONTH(f.fecha_emision_factura) AS mes,
  SUM(f.total_factura) AS ventas_totales
FROM
  facturas f
GROUP BY
  YEAR(f.fecha_emision_factura), MONTH(f.fecha_emision_factura);
```

año	mes	ventas_totales
2024	9	3867.50

3. Promedio de Ventas por Cliente

```
SELECT
    c.nombre_cliente,
    AVG(f.total_factura) AS promedio_ventas
FROM
    clientes c
JOIN
    facturas f ON c.id_cliente = f.id_cliente
GROUP BY
    c.nombre_cliente;
```

% <	
Results Messages	
nombre_cliente	promedio_ventas
Ciente 1	119.000000
Ciente 10	654.500000
Ciente 2	178.500000
Ciente 3	238.000000
Ciente 4	297.500000

4. Productos más Vendidos

```
SELECT
    p.nombre_producto,
    COALESCE(SUM(op.cantidad), 0) AS total_vendido
FROM
    productos p
LEFT JOIN
    ordenes_productos op ON p.id_producto = op.id_producto
GROUP BY
    p.nombre_producto
ORDER BY
    total_vendido DESC;
```

% <	
Results Messages	
nombre_producto	total_vendido
Bolsos de Mano	0
Bufanda de Lana	0
Calcetines de Algodón	0

5. Reporte de Productos: Stock Actual y Niveles Mínimos

```
SELECT
    p.nombre_producto,
    i.stock_actual,
    a.nivel_minimo
FROM
    productos p
JOIN
    inventario i ON p.id_producto = i.id_producto
LEFT JOIN
    alertas_stock a ON p.id_producto = a.id_producto;
```

nombre_producto	stock_actual	nivel_minimo
Producto A	40	5
Producto A	40	5
Producto B	25	5
Producto B	25	10

6. Cantidad de Empleados por Tienda

```
SELECT
    t.nombre_tienda,
    COUNT(e.id_empleado) AS total_empleados
FROM
    tiendas t
LEFT JOIN
    empleados e ON t.id_tienda = e.id_tienda
GROUP BY
    t.nombre_tienda;
```

nombre_tienda	total_empleados
Tienda Bogotá	2
Tienda Buenos Aires	2
Tienda Cali	2
Tienda Centro	1

7. Inventario Total por Subcategoría

```
SELECT
    sc.nombre_sub_categoria,
    SUM(i.stock_actual) AS stock_total
FROM
    sub_categorias sc
JOIN
    productos p ON sc.id_sub_categoria = p.id_sub_categoria
JOIN
    inventario i ON p.id_producto = i.id_producto
GROUP BY
    sc.nombre_sub_categoria;
```

nombre_sub_categoria	stock_total
Camisetas	15
Celulares	40
Computadoras	25
Gorras	85
Juguetes educativos	95

8. Número Total de Proveedores por Ciudad

```
SELECT
    c.nombre_ciudad,
    COUNT(DISTINCT p.id_proveedor) AS total_proveedores
FROM
    ciudades c
LEFT JOIN
    proveedores p ON c.id_ciudad = p.id_ciudad
GROUP BY
    c.nombre_ciudad;
```

nombre_ciudad	total_proveedores
Arequipa	2
Bello	2
Bogotá	3
Buenos Aires	3
Cali	3

9. Valor del Inventario:

Esta consulta calcula el valor total del inventario disponible.

```
SELECT
    SUM(p.precio_producto * i.stock_actual) AS Valor_Inventario
FROM
    productos p
JOIN
    inventario i ON p.id_producto = i.id_producto;
```

Results	Messages
Valor_Inventario	
40500.00	

10. Promedio de Tiempo de Almacenamiento:

Calcula el tiempo promedio que un producto permanece en el inventario antes de ser vendido.

```
SELECT
    AVG(DATEDIFF(DAY, h.fecha_cambio, GETDATE())) AS Promedio_Tiempo
FROM
    historial_inventario h;
```

Results	Messages
Promedio_Tiempo	
13	

11. Porcentaje de Productos por Categoría

Consulta el porcentaje de productos por cada categoría en relación con el total de productos.

```
SELECT
    c.nombre_categoria,
    COUNT(p.id_producto) * 1.0 / (SELECT COUNT(*) FROM productos) * 100 AS Porcentaje
FROM
    categorias c
LEFT JOIN
    sub_categorias sc ON c.id_categoria = sc.id_categoria
LEFT JOIN
    productos p ON sc.id_sub_categoria = p.id_sub_categoria
GROUP BY
    c.nombre_categoria;
```

sults Messages	
nombre_categoria	Porcentaje
Electrónica	56.666666666...
Guantes	0.000000000000
Hogar	0.000000000000
Juguetes	3.333333333300

12. Margen de Ganancia Bruto:

Determina el margen de ganancia bruto de la empresa.

```
SELECT
    (SUM(f.total_factura) - SUM(f.sub_total_factura))
    / SUM(f.total_factura) * 100 AS Margen_Bruto
FROM
    facturas f;
```

sults Messages	
Margen_Bruto	
15.966300	

13. Número de Proveedores Activos:

Consulta el número de proveedores activos que tienen un correo electrónico registrado (asumiendo que esto significa que están activos).

```
SELECT
    COUNT(DISTINCT p.id_proveedor) AS Proveedores_Activos
FROM
    proveedores p
WHERE
    p.email_proveedor IS NOT NULL;
```

Results	Messages
Proveedores_Activos	
50	

14. Promedio de Facturación por Cliente:

Calcula el promedio de facturación por cliente basado en las facturas registradas.

```
SELECT
    SUM(f.total_factura) * 1.0 / COUNT(DISTINCT f.id_cliente)
    AS Promedio_Facturacion
FROM
    facturas f;
```

Results	Messages
Promedio_Facturacion	
386.750000	

15. Tasa de Disponibilidad de Productos:

Mide el porcentaje de productos que están disponibles en stock en comparación con el total de productos.

```
SELECT
    (CAST(COUNT(DISTINCT p.id_producto) AS FLOAT) / (SELECT COUNT(*) FROM productos))
    * 100 AS Tasa_Disponibilidad
FROM
    inventario i
JOIN
    productos p ON i.id_producto = p.id_producto
WHERE
    i.stock_actual > 0;
```

Results	Messages
Tasa_Disponibilidad	
33.333333333333	

16. Productos más Vendidos por Categoría:

Identifica los productos más vendidos en cada categoría con base en la cantidad de pedidos.

```
SELECT
    c.nombre_categoria,
    p.nombre_producto,
    SUM(op.cantidad) AS Total_Vendido
FROM
    ordenes_productos op
JOIN
    productos p ON op.id_producto = p.id_producto
JOIN
    sub_categorias sc ON p.id_sub_categoria = sc.id_sub_categoria
JOIN
    categorias c ON sc.id_categoria = c.id_categoria
GROUP BY
    c.nombre_categoria, p.nombre_producto
ORDER BY
    c.nombre_categoria, Total_Vendido DESC;
```

nombre_categoria	nombre_producto	Total_Vendido
Electrónica	Vestido de Verano	76
Electrónica	Chaqueta de Invierno	6
Electrónica	Calcetines de Algodón	5

17. Ingresos Totales por Tienda:

Consulta los ingresos generados por cada tienda con base en las facturas registradas.

```
SELECT
    t.nombre_tienda,
    SUM(f.total_factura) AS Ingresos_Totales
FROM
    facturas f
JOIN
    productos p ON f.id_factura = p.id_factura
JOIN
    tiendas t ON p.id_tienda = t.id_tienda
GROUP BY
    t.nombre_tienda
ORDER BY
    Ingresos_Totales DESC;
```

nombre_tienda	Ingresos_Totales
Tienda Bogotá	773.50
Tienda Quito	654.50
Tienda Lima	595.00
Tienda Río de Janeiro	535.50
Tienda Buenos Aires	476.00
Tienda Monterrey	416.50
Tienda San Francisco	357.00
Tienda El Paso	297.50
Tienda Cali	238.00
Tienda Medellín	119.00

18. Clientes con Mayor Número de Compras:

Identifica los clientes con mayor número de facturas emitidas.

```
SELECT
    c.nombre_cliente,
    c.email_cliente,
    COUNT(f.id_factura) AS Numero_Compras
FROM
    facturas f
JOIN
    clientes c ON f.id_cliente = c.id_cliente
GROUP BY
    c.nombre_cliente, c.email_cliente
ORDER BY
    Numero_Compras DESC;
```

% ▾		
Results Messages		
nombre_cliente	email_cliente	Numero_Compras
Cliente 1	cliente1@correo.com	1
Cliente 10	cliente10@correo.com	1

19. Porcentaje de Métodos de Pago más Utilizados:

Consulta la frecuencia del método de pago más usado y calcula el porcentaje.

```
SELECT TOP 1
    metodo_pago,
    COUNT(id_metodo_pago) AS Cantidad_Usos,
    COUNT(id_metodo_pago) * 100.0 / (SELECT COUNT(*) FROM metodos_de_pago) AS Porcentaje_Utilizacion
FROM
    metodos_de_pago
GROUP BY
    metodo_pago
ORDER BY
    Cantidad_Usos DESC;
```

Results Messages		
metodo_pago	Cantidad_Usos	Porcentaje_Utilizacion
EFFECTIVO	3	30.000000000000

20. Rendimiento de Venta por Empleado:

Mide el rendimiento de cada empleado basándose en las ventas que ha generado.

```
SELECT
    e.nombres_empleado + ' ' + e.apellidos_empleado AS Empleado,
    SUM(f.total_factura) AS Ventas_Generadas
FROM
    empleados e
JOIN
    ordenes_de_compra o ON e.id_empleado = o.id_empleado
JOIN
    facturas f ON o.id_factura = f.id_factura
GROUP BY
    e.nombres_empleado, e.apellidos_empleado
ORDER BY
    Ventas_Generadas DESC;
```

Results Messages	
Empleado	Ventas_Generadas
Jorge Paredes	654.50
Sofía Díaz	595.00
Andrés Ramírez	535.50
Patricia García	476.00
José Hernández	416.50
María López	357.00
Carlos Martínez	297.50
Luis Rodríguez	238.00
Ana Gómez	178.50
Santiago Pérez	119.00