

SQL

DDL – DML – DQL

BASE DE DATOS: db_empresa

TABLA: **inventarios**

MODELO FISICO

Nombre del Campo	Tipo de Dato	Restricciones
idInventario	INT	PRIMARY KEY, AUTO_INCREMENT
codigoProducto	VARCHAR(50)	NOT NULL, UNIQUE
nombre	VARCHAR(100)	NOT NULL
descripcion	TEXT	NULL
fechaFabricacion	DATE	NULL
fechaRegistro	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP
cantidadProducto	INT	DEFAULT 0
precioCompra	DECIMAL(10,2)	NOT NULL
precioVenta	DECIMAL(10,2)	NOT NULL

DDL

CREATE - DATABASE - TABLE



```
1  CREATE DATABASE IF NOT EXISTS db_empresa;  
2  USE db_empresa;  
3  
4  CREATE TABLE IF NOT EXISTS inventarios (  
5      idInventario INT PRIMARY KEY AUTO_INCREMENT,  
6      codigoProducto VARCHAR(50) NOT NULL UNIQUE,  
7      nombre VARCHAR(100) NOT NULL,  
8      descripcion TEXT,  
9      fechaFabricacion DATE,  
10     fechaRegistro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
11     cantidadProducto INT DEFAULT 0,  
12     precioCompra DECIMAL(10,2) NOT NULL,  
13     precioVenta DECIMAL(10,2) NOT NULL  
14 );  
15
```

DDL

- CREATE
- DATABASE
- TABLE

DML

CRUD

INSERT INTO - UPDATE - DELETE



```
1  INSERT INTO inventarios (codigoProducto,
2                             nombre,
3                             descripcion,
4                             fechaFabricacion,
5                             cantidadProducto,
6                             precioCompra,
7                             precioVenta)
8  VALUES ('PROD001',
9           'Laptop HP',
10          'Laptop HP con procesador Intel i7, 16GB RAM',
11          '2024-01-10',
12          10,
13          800.00,
14          1200.00);
```

DML

• INSERT INTO

Paso 1: Inserción de Datos

- Agreguemos un producto al inventario

Sintaxis:

```
INSERT INTO table_name (column_a, column_b)
VALUES ("value_a", "value_b");
```

INSERCIÓN MULTIPLE



```
1  INSERT INTO inventarios (codigoProducto,
2                             nombre,
3                             descripcion,
4                             fechaFabricacion,
5                             cantidadProducto,
6                             precioCompra,
7                             precioVenta)
8  VALUES
9  ('PROD002', 'Monitor Samsung 24"', 'Monitor LED Samsung de 24 pulgadas', '2024-02-15', 25, 120.00, 180.00),
10 ('PROD003', 'Mouse Logitech', 'Mouse inalámbrico Logitech', '2024-03-01', 50, 15.00, 30.00),
11 ('PROD004', 'Teclado Mecánico', 'Teclado mecánico retroiluminado RGB', '2024-03-10', 40, 60.00, 100.00);
12
13
```



```
1  /*Imagina que recibes un lote adicional de
2  20 laptops y necesitas actualizar la cantidad:*/
3  UPDATE inventarios
4  SET cantidadProducto = cantidadProducto + 20
5  WHERE codigoProducto = 'PROD001';
6
7  /*Supón que decides aumentar el precio de venta
8  de todos los productos en un 10%:*/
9  UPDATE inventarios
10 SET precioVenta = precioVenta * 1.10;
```

DML

• UPDATE

Paso 2: Actualización de Datos

- Actualizar la cantidad de un producto
- Modificar el precio de venta de un producto



```
1  /*Digamos que decides retirar un producto
2     que ya no vendes más:*/
3  DELETE FROM inventarios
4  WHERE codigoProducto = 'PROD004';
5
6  /*Supón que quieres limpiar tu inventario
7     de productos agotados:*/
8  DELETE FROM inventarios
9  WHERE cantidadProducto = 0;
10
```

DML

- **DELETE**

Paso 3: Eliminación de Datos

- Aprender a eliminar registros específicos sin afectar toda la tabla.

DQL

Lenguaje de Consulta de Datos

SELECT - WHERE - ORDER BY

FUNCIONES DE AGREGACIÓN

GROUP BY - HAVING

OBJETIVOS:

- Familiarizarse con la selección básica de datos.
- Filtrar datos basados en valores específicos.
- Usar patrones de búsqueda con LIKE.
- Ordenar resultados con ORDER BY.
- Utilizar subconsultas para encontrar valores extremos (MAX, MIN).
- Aplicar funciones de agregación como SUM y AVG.
- Agrupar datos usando GROUP BY.
- Trabajar con fechas y filtrar registros recientes.



```
1  -- Selecciona todos los campos (*) de la tabla 'inventarios' y los muestra
2  SELECT * FROM inventarios;
3
4  -- Selecciona campos especificos de la tabla 'inventarios' y los muestra
5  SELECT nombre, cantidadProducto FROM inventarios;
6
7  -- Selecciona todos los campos (*) de la tabla 'inventarios'
8  -- donde la cantidad de producto (cantidadProducto) es mayor a 20.
9  SELECT * FROM inventarios WHERE cantidadProducto > 10;
10
11 -- Selecciona todos los campos (*) de la tabla 'inventarios'
12 -- donde el nombre del producto contiene la palabra "Monitor".
13 SELECT * FROM inventarios WHERE nombre LIKE '%Monitor%';
14
```

DQL

• SELECT

Paso 4: Consultas básicas

- Selecciona la table y realiza las filtraciones.

DQL

- **ORDER BY**

Paso 5: Ordena Resultados

- Selecciona la table y realiza ordenación de datos con ASC , DESC.

```
1  -- Selecciona todos los campos de la tabla 'inventarios'
2  -- y los ordena de forma descendente (DESC) por el campo 'fechaFabricacion'.
3  SELECT * FROM inventarios ORDER BY fechaFabricacion DESC;
```



```
1  -- Selecciona cuenta todos los registros de la tabla 'inventarios'
2  SELECT COUNT(*) FROM inventarios;
3
4  -- Utiliza la función de agregación AVG para calcular el precio promedio
5  -- de todos los productos en el campo 'precioVenta'
6  SELECT AVG(precioVenta) FROM inventarios;
7
8  -- Selecciona todos los campos de la tabla 'inventarios'
9  -- donde el precio de venta es mayor a 100 y la fecha de
10 -- fabricación es posterior al 1 de enero de 2024.
11 SELECT * FROM inventarios
12 WHERE precioVenta > 100 AND fechaFabricacion > '2024-01-01';
13
14 -- Utiliza la función de agregación MIN para encontrar el precio de compra más bajo,
15 -- y selecciona todas los campos para ese producto específico.
16 SELECT * FROM inventarios WHERE precioCompra = (SELECT MIN(precioCompra) FROM inventarios);
17
18 -- Utiliza la función de agregación SUM para calcular la cantidad total
19 -- de productos en el campo 'cantidadProducto'.
20 SELECT SUM(cantidadProducto) AS totalProductos FROM inventarios;
21
```

DQL

• FUNCIONES

Paso 6: FUNCIONES

- Operaciones matemáticas.



```
1  -- Agrupar productos por nombre y mostrar la cantidad total disponible para cada grupo
2  SELECT nombre, SUM(cantidadProducto) AS totalPorNombre
3  FROM inventarios
4  GROUP BY nombre;
5
6  -- Mostrar solo aquellos grupos donde la cantidad total sea mayor a 30
7  SELECT nombre, SUM(cantidadProducto) AS totalPorNombre
8  FROM inventarios
9  GROUP BY nombre
10 HAVING SUM(cantidadProducto) > 30;
```

DQL

- **GROUP BY**

Paso 7: AGRUPACIONES

- Combinación de instrucciones.

EJERCICIOS CONSULTAS BÁSICA , ORDENAMIENTO Y AGREGACIÓN



```
1  -- Ejercicios - CONSULTAS BÁSICA , ORDENAMIENTO Y AGREGACIÓN
2  --1. Listar todos los productos en inventario ( * )
3  --2. Mostrar productos cuyo precio de venta sea superior a $200. ( > , < , >= , <= , = , != )
4  --3. Mostrar el nombre y la cantidad de productos cuyo código de producto comienza con 'A'. ('A%')
5  --4. Ordenar productos por precio de compra en orden ascendente. (ASC)
6  --5. Encontrar el producto más caro. (MAX)
7  --6. Calcular el total de productos disponibles en el inventario. (SUM)
8  --7. Agrupar productos por fecha de fabricación y mostrar la cantidad total de productos fabricados por cada fecha ( SUM )
9  --8. Mostrar el producto con la cantidad más baja en inventario ( MIN )
10 --9. Listar productos registrados en los últimos 30 días
11 --10. Calcular el precio promedio de compra de los productos que tienen más de 10 unidades disponibles
12
```


