WorkShop 3

Preparando el ambiente

Entramos a Mysql `sudo mysql`

Creamos la DB `create database northwind`

Creamos un usuario `create user user_laravel identified by "12345"`

Le damos permisos `grant all privileges on northwind.\* to user_laravel`

Habilitamos las conexiones remotas de mysql `nano /etc/mysql/mariadb.conf.d/50-server.cnf`

Y comentamos la linea

```
# bind-address = 127.0.0.1
```

Reiniciamos mariadb `sudo systemctl restart mysql`

Cargamos nuestro backup `mysql northwind > northwind.sql mysql northwind > northwind-data.sql`

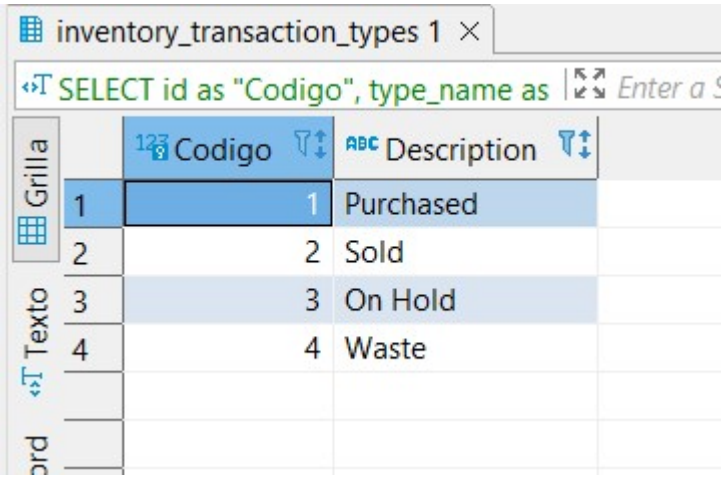
Nos conectamos desde la maquina local con un gestor como DBEAVER o TABLEPLUS

Challenges

Challenge 1

Recupere el código (id) y la descripción (type_name) de los tipos de movimiento de inventario (inventory_transaction_types).

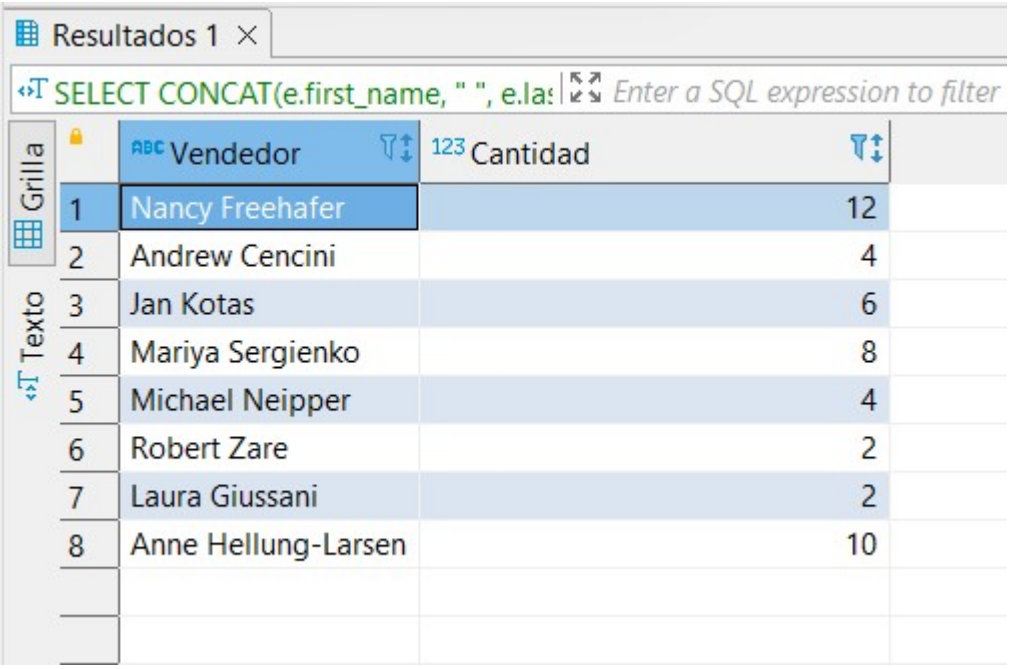
```
SELECT id as "Codigo", type_name as "Description" FROM inventory_transaction_types
```



Challenge 2

Recupere la cantidad de ordenes (orders) registradas por cada vendedor (employees).

```
SELECT
CONCAT(e.first_name, " ", e.last_name) as `Vendedor`,
COUNT(1) as `Cantidad`
from orders o
join employees e
on e.id = o.employee_id
GROUP BY `Vendedor`
ORDER BY e.id
```



Challenge 3

Recupere la lista de los 10 productos más ordenados (order_details), y la cantidad total de unidades ordenadas para cada uno de los productos. Deberá incluir como mínimo los campos de código (product_code), nombre del producto (product_name) y la cantidad de unidades.

```
select p.product_code as `Codigo`,
        p.product_name as `Producto`,
        round(sum(od.quantity),2) as `Cantidad`
        from products p
        join order_details od
            on od.product_id = p.id
        group by p.id
        order by `Cantidad` DESC
        limit 10;
```

ct p.product_ Enter a SQL expression to filter results (use Ctrl+Space)

Codigo	Producto	Cantidad
NWTB-43	Northwind Traders Coffee	650
NWTB-34	Northwind Traders Beer	487
NWTSO-41	Northwind Traders Clam Chowder	290
NWTB-81	Northwind Traders Green Tea	275
NWTCA-48	Northwind Traders Chocolate	200
NWTCM-40	Northwind Traders Crab Meat	120
NWTJP-6	Northwind Traders Boysenberry Spread	100
NWTP-57	Northwind Traders Ravioli	100
NWTD-72	Northwind Traders Mozzarella	90
NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	85

Challenge 4

Recupere el monto total (invoices, orders, order_details, products) y la cantidad de facturas (invoices) por vendedor (employee). Debe considerar solamente las ordenes con estado diferente de 0 y solamente los detalles en estado 2 y 3, debe utilizar el precio unitario de las lineas de detalle de orden, no considere el descuento, no considere los impuestos, porque la comisión a los vendedores se paga sobre el precio base.

```
select count(1) as `Cantidad`,
        concat(e.first_name, ' ', e.last_name) as `Vendedor`,
        round(sum(od.unit_price * od.quantity),2) as `Monto`
        from employees e
        join orders o
            on o.employee_id = e.id
        join order_details od
            on od.order_id = o.id
        where o.status_id <> 0
            and od.status_id in (2,3)
        group by e.id
        order by `Cantidad` DESC, e.first_name ASC;
```

	Cantidad	Vendedor	Monto
1	10	Anne Hellung-Larsen	19.974,25
2	10	Mariya Sergienko	6.278
3	8	Nancy Freehafer	4.972,5
4	7	Jan Kotas	5.787,5
5	4	Michael Neipper	6.378
6	3	Andrew Cencini	2.617,5
7	3	Robert Zare	3.786,5
8	1	Laura Giussani	680

Challenge 5

Recupere el monto total (invoices, orders, order_details, products) y la cantidad de facturas (invoices) por vendedor (employee). Debe considerar solamente las ordenes con estado diferente de 0 y solamente los detalles en estado 2 y 3, debe utilizar el precio unitario de las lineas de detalle de orden, no considere el descuento, no considere los impuestos, porque la comisión a los vendedores se paga sobre el precio base.

```
select p.product_code as `Codigo`,
    p.product_name as `Producto`,
    sum(it.quantity) as `Cantidad`
    from products p
    join inventory_transactions it
        on it.product_id = p.id
    where it.transaction_type in (1)
    group by p.id;
```

products 1

select p.product_cod Enter a SQL expression to filter re

Grilla

Texto

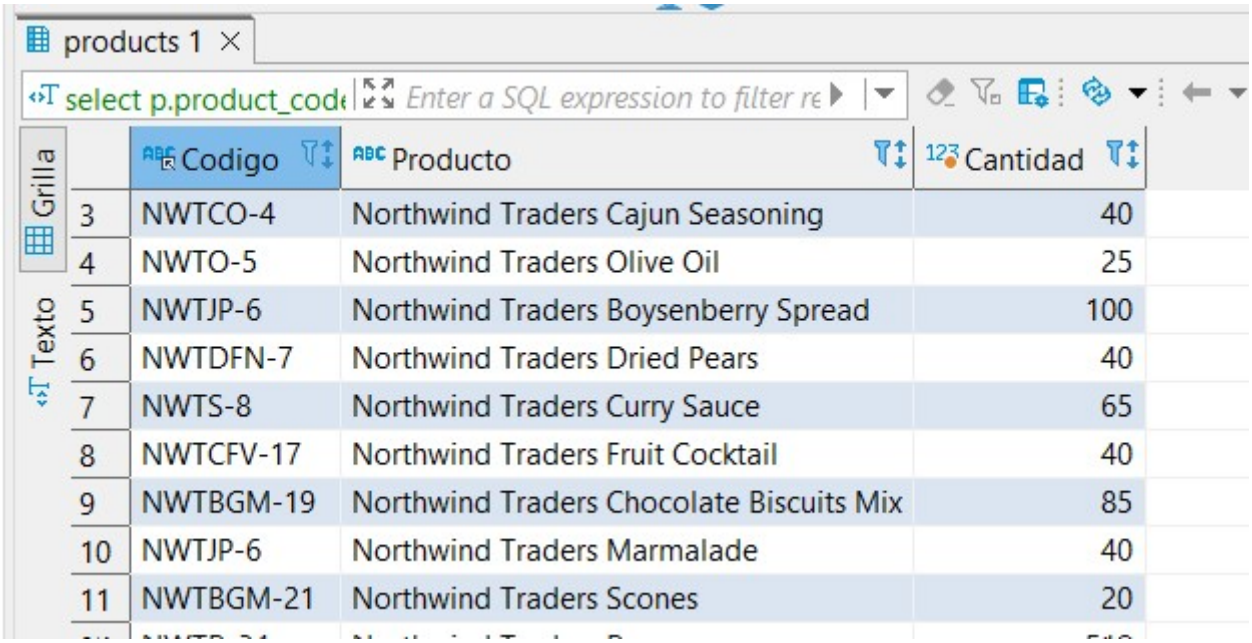
	Codigo	Producto	Cantidad
3	NWTCO-4	Northwind Traders Cajun Seasoning	40
4	NWTO-5	Northwind Traders Olive Oil	40
5	NWTJP-6	Northwind Traders Boysenberry Spread	100
6	NWTDFN-7	Northwind Traders Dried Pears	40
7	NWTS-8	Northwind Traders Curry Sauce	65
8	NWTDFN-14	Northwind Traders Walnuts	40
9	NWTCFV-17	Northwind Traders Fruit Cocktail	40
10	NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	85

Challenge 6

Recupere los movimientos de inventario del tipo salida. Tomando como base todos los movimientos de inventario (inventory_transactions), considere unicamente los tipos de movimiento (transaction_type) 2, 3 y 4

como salidas. Debe agrupar por producto (products) y deberá incluir como mínimo los campos de código (product_code), nombre del producto (product_name) y la cantidad de unidades que salieron.

```
select p.product_code as `Codigo`,
    p.product_name as `Producto`,
    sum(it.quantity) as `Cantidad`
    from products p
    join inventory_transactions it
        on it.product_id = p.id
    where it.transaction_type in (2,3,4)
    group by p.id;
```



products 1

select p.product_cod | Enter a SQL expression to filter re

Grilla | Texto

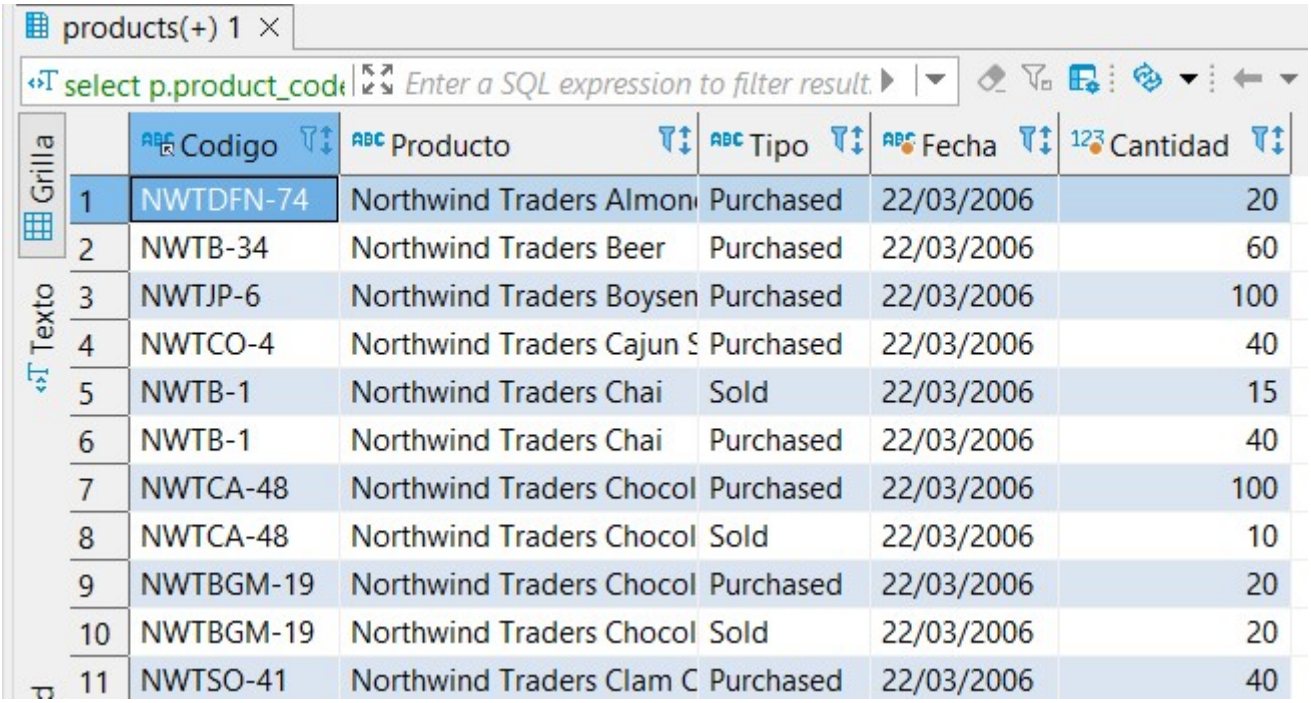
	Codigo	Producto	Cantidad
3	NWTCO-4	Northwind Traders Cajun Seasoning	40
4	NWTO-5	Northwind Traders Olive Oil	25
5	NWTJP-6	Northwind Traders Boysenberry Spread	100
6	NWTDFN-7	Northwind Traders Dried Pears	40
7	NWTS-8	Northwind Traders Curry Sauce	65
8	NWTCFV-17	Northwind Traders Fruit Cocktail	40
9	NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	85
10	NWTJP-6	Northwind Traders Marmalade	40
11	NWTBGM-21	Northwind Traders Scones	20

Challenge 7

Genere un reporte de movimientos de inventario (inventory_transactions) por producto (products), tipo de transacción y fecha, entre las fechas 22/03/2006 y 24/03/2006 (incluyendo ambas fechas). Debe incluir como mínimo el código (product_code), el nombre del producto (product_name), la fecha truncada (transaction_created_date), la descripción del tipo de movimiento (type name) y la suma de cantidad (quantity) .

```
select p.product_code as `Codigo`,
    p.product_name as `Producto`,
    itt.type_name as `Tipo`,
    date_format(it.transaction_created_date, "%d/%m/%Y") as `Fecha`,
    sum(it.quantity) as `Cantidad`
    from inventory_transactions it
    join products p
        on it.product_id = p.id
    join inventory_transaction_types itt
        on it.transaction_type = itt.id
    where it.transaction_created_date between '2006-03-22' and '2006-03-24'
```

```
    group by it.id
    order by p.product_name asc;
```



products(+) 1

select p.product_code Enter a SQL expression to filter result

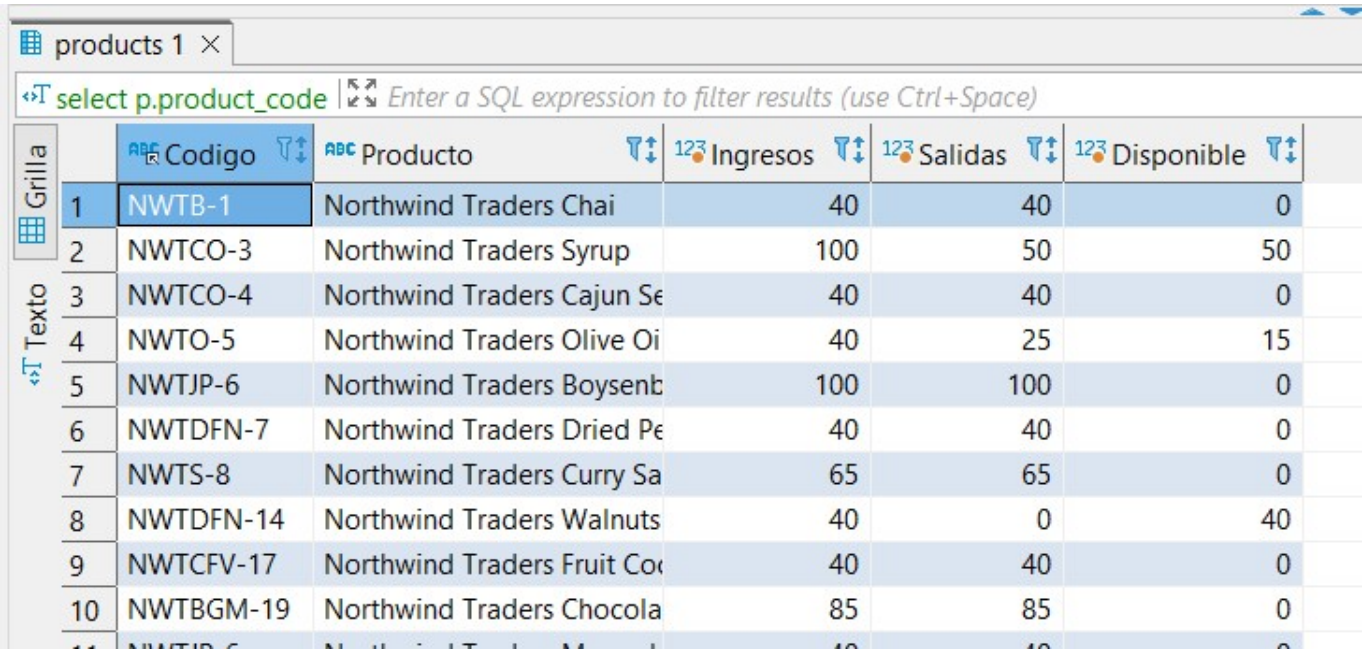
Grilla Texto

	Codigo	Producto	Tipo	Fecha	Cantidad
1	NWTDFN-74	Northwind Traders Almon	Purchased	22/03/2006	20
2	NWTB-34	Northwind Traders Beer	Purchased	22/03/2006	60
3	NWTJP-6	Northwind Traders Boysen	Purchased	22/03/2006	100
4	NWTCO-4	Northwind Traders Cajun S	Purchased	22/03/2006	40
5	NWTB-1	Northwind Traders Chai	Sold	22/03/2006	15
6	NWTB-1	Northwind Traders Chai	Purchased	22/03/2006	40
7	NWTCA-48	Northwind Traders Chocol	Purchased	22/03/2006	100
8	NWTCA-48	Northwind Traders Chocol	Sold	22/03/2006	10
9	NWTBGM-19	Northwind Traders Chocol	Purchased	22/03/2006	20
10	NWTBGM-19	Northwind Traders Chocol	Sold	22/03/2006	20
11	NWTSO-41	Northwind Traders Clam C	Purchased	22/03/2006	40

Challenge 8

Genere la consulta SQL para un reporte de inventario, tomando como base todos los movimientos de inventario (inventory_transactions), considere los tipos de movimiento (transaction_type) 2, 3 y 4 como salidas y el tipo 1 como ingreso. Este reporte debe estar agrupado por producto (products) y deberá incluir como mínimo los campos de código (product_code), nombre del producto (product_name) y la sumarización de ingresos, salidas y la cantidad disponible en inventario (diferencia de ingresos - salidas).

```
select p.product_code as `Codigo`,
    p.product_name as `Producto`,
    sum(if(it.transaction_type = 1, it.quantity, 0)) as `Ingresos`,
    sum(if(it.transaction_type > 1, it.quantity, 0)) as `Salidas`,
    sum(if(it.transaction_type = 1, it.quantity, 0)) - sum(if(it.transaction_type
> 1, it.quantity, 0)) as `Disponible`
    from products p
    join inventory_transactions it
        on it.product_id = p.id
    group by p.id;
```



products 1

select p.product_code | Enter a SQL expression to filter results (use Ctrl+Space)

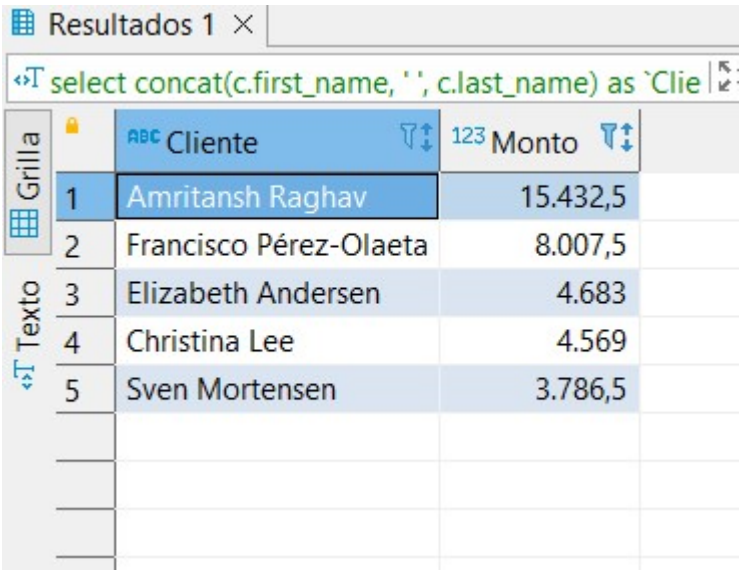
Grilla | Texto

	Codigo	Producto	Ingresos	Salidas	Disponible
1	NWTB-1	Northwind Traders Chai	40	40	0
2	NWTCO-3	Northwind Traders Syrup	100	50	50
3	NWTCO-4	Northwind Traders Cajun Se	40	40	0
4	NWTO-5	Northwind Traders Olive Oi	40	25	15
5	NWTJP-6	Northwind Traders Boysenb	100	100	0
6	NWTDFN-7	Northwind Traders Dried Pe	40	40	0
7	NWTS-8	Northwind Traders Curry Sa	65	65	0
8	NWTDFN-14	Northwind Traders Walnuts	40	0	40
9	NWTCFV-17	Northwind Traders Fruit Co	40	40	0
10	NWTBGM-19	Northwind Traders Chocola	85	85	0

Challenge 9 - Extra

Determinar el mejor cliente

```
select concat(c.first_name, ' ', c.last_name) as `Cliente`,
    round(sum(od.quantity * od.unit_price), 2) as `Monto`
    from customers c
    join orders o
        on o.customer_id = c.id
    join order_details od
        on od.order_id = o.id
    join invoices i
        on i.order_id = o.id
    group by c.id
    order by `Monto` desc
    limit 5;
```



Resultados 1

select concat(c.first_name, ' ', c.last_name) as `Clie

Grilla | Texto

	Cliente	Monto
1	Amritansh Raghav	15.432,5
2	Francisco Pérez-Olaeta	8.007,5
3	Elizabeth Andersen	4.683
4	Christina Lee	4.569
5	Sven Mortensen	3.786,5

Challenge 10 - Extra

Determinar el mejor mes

```
select date_format(i.invoice_date, "%Y") as `Ano`,
    date_format(i.invoice_date, "%m") as `Mes`,
    round(sum(od.quantity * od.unit_price), 2) as `Monto`
    from invoices i
    join orders o
        on o.id = i.order_id
    join order_details od
        on od.order_id = o.id
    group by `Ano`, `Mes`
    order by `Monto` desc;
```