-- Funciones de agregacion

-- el importe de ventas total

select sum(quantity \* unitprice) TotalVentas

from [Order Details]

select ProductId, sum(quantity \* unitprice) TotalVentas

from [Order Details]

group by ProductId



-- count

select \* from Orders

select count('Hola') from Orders -- 830

select count(1) from Orders -- 830

select customerId, count(1) NroPedidos

from Orders

group by customerId



select customerId, count(1) NroPedidos

from Orders

where OrderDate between '2020-01-01' and '2020-12-31'

group by customerId

select customerId, count(1) NroPedidos

from Orders

where OrderDate between '2020-01-01' and '2020-12-31'

and count(1) = 3

group by customerId

-- An aggregate may not appear in the WHERE clause unless it is in a subquery contained

-- in a HAVING clause or a select list, and the column being aggregated is an outer reference.



select customerId, count(1) NroPedidos

from Orders

where OrderDate between '2020-01-01' and '2020-12-31'

group by customerId

having count(1) > 3

-- la 1ra compra y cuando fue la ultima compra

select \* from Orders

select CustomerId, min(OrderDate) [1ra Compra], max(OrderDate) [Ult Compra]

from Orders

group by CustomerID

-- Obtener el detalle de pedidos que supere el promedio de la cantidad

select \* from [Order Details]



select avg(quantity) from [Order Details]

-- Devuelve un valor escalar: una columna y una fila:

-- Opcion #1 Constante

select \* from [Order Details] where Quantity > 23

-- Opcion #2 Variable

declare @Promedio int -- para promedio decimal, declararla como money

select @promedio = avg(quantity) from [Order Details]

select \* from [Order Details] where Quantity > @Promedio

-- obtener un escalar (1f,1c) y pasarlo a variable

-- no se usa set porque lo que se busca es leer una tabla

-- ambas opciones arrojan el mismo resultado: