Lab 2 - Funções, Stored Procedures, Triggers e Metadados

Nome: Nuno Reis Número: [202000753](https://portal.ips.pt/ests/pt/fest_geral.cursos_list?pv_num_unico=202000753)

Etapa 1

--a)

create or alter function dbo.fnTotalVendasProduto (@product\_id int)

returns int

AS

begin

declare @total int;

select @total=sum(OrderQty\*UnitPrice)

from SalesLT.SalesOrderDetail

where ProductID = @product\_id;

return @total;

end;

select dbo.fnTotalVendasProduto(976) as Total;

--b)

select Name, dbo.fnTotalVendasProduto(ProductID) as Total

from SalesLT.Product

where dbo.fnTotalVendasProduto(ProductID) is not null;

--c)

create or alter function dbo.fnTotalVendas ()

returns int

as

begin

declare @total int =0;

select @total += dbo.fnTotalVendasProduto(ProductID)

from SalesLT.Product

where dbo.fnTotalVendasProduto(ProductID) is not null

return @total;

end;

select dbo.fnTotalVendas() as Total;

--d)

create or alter procedure dbo.spClientesCidade @city nvarchar(30)

as

select \*

from SalesLT.Address

where SalesLT.Address.City=@City;

go

exec dbo.spClientesCidade @City = "Las Vegas";

--e)

create or alter procedure dbo.spListaCompra @SalesOID int

as

declare @costumerEmail nvarchar(50);

declare @orderID int;

declare @date datetime;

declare @total int;

--Costumer

declare CostumerEmail\_Cursor cursor

for select c.EmailAddress

from [SalesLT].[SalesOrderHeader] as h

left join [SalesLT].[Customer] as c

on h.CustomerID=c.CustomerID

where h.SalesOrderID=@SalesOID;

open CostumerEmail\_Cursor

fetch next from CostumerEmail\_Cursor into @costumerEmail;

close CostumerEmail\_Cursor;

deallocate CostumerEmail\_Cursor;

--Order

declare OrderID\_Cursor cursor

for select h.SalesOrderID

from [SalesLT].[SalesOrderHeader] as h

where h.SalesOrderID=@SalesOID;

open OrderID\_Cursor

fetch next from OrderID\_Cursor into @orderID;

close OrderID\_Cursor;

deallocate OrderID\_Cursor;

--Date

declare Date\_Cursor cursor

for select h.OrderDate

from [SalesLT].[SalesOrderHeader] as h

where h.SalesOrderID=@SalesOID;

open Date\_Cursor

fetch next from Date\_Cursor into @date;

close Date\_Cursor;

deallocate Date\_Cursor;

--Total

declare Total\_Cursor cursor

for select sum(OrderQty\*UnitPrice)

from SalesLT.SalesOrderDetail as h

where h.SalesOrderID=@SalesOID;

open Total\_Cursor

fetch next from Total\_Cursor into @total;

close Total\_Cursor;

deallocate Total\_Cursor;

--Print do cabeçalho

print '--------------------------------------------'

print 'Costumer: ' + @costumerEmail;

print 'Order: SO' + cast(@orderID as varchar(10));

print 'Date: ' + CONVERT( VARCHAR(24), @date, 121);

print 'Total: ' + cast(@total as varchar(10));

--Print da lista de produtos

print '--------------------------------------------'

declare @salesCount int;

set @salesCount = (select COUNT(\*)

from [SalesLT].[SalesOrderDetail] as d

where d.SalesOrderID = @SalesOID

group by d.SalesOrderID);

while @salesCount > 0

begin

declare @productName nvarchar(50);

declare @saleOrderQty int;

declare @saleUnitPrice float;

declare @saleUnitPriceDiscount float;

declare @saleLineTotal float;

--Product name

declare ProductName\_Cursor cursor

for select p.Name--

from [SalesLT].[SalesOrderDetail] as d

left join [SalesLT].Product as p

on d.ProductID=p.ProductID

where d.SalesOrderID=@SalesOID;

open ProductName\_Cursor

fetch next from ProductName\_Cursor into @productName;

--Sale Order Qty

declare SaleOrderQty\_Cursor cursor

for select d.OrderQty

from [SalesLT].[SalesOrderDetail] as d

left join [SalesLT].Product as p

on d.ProductID=p.ProductID

where d.SalesOrderID=@SalesOID;

open SaleOrderQty\_Cursor

fetch next from SaleOrderQty\_Cursor into @saleOrderQty;

--Sale Unit Price

declare SaleUnitPrice\_Cursor cursor

for select d.UnitPrice

from [SalesLT].[SalesOrderDetail] as d

left join [SalesLT].Product as p

on d.ProductID=p.ProductID

where d.SalesOrderID=@SalesOID;

open SaleUnitPrice\_Cursor

fetch next from SaleUnitPrice\_Cursor into @saleUnitPrice;

--Sale Unit Price Discount

declare SaleUnitPriceDiscount\_Cursor cursor

for select d.UnitPriceDiscount

from [SalesLT].[SalesOrderDetail] as d

left join [SalesLT].Product as p

on d.ProductID=p.ProductID

where d.SalesOrderID=@SalesOID;

open SaleUnitPriceDiscount\_Cursor

fetch next from SaleUnitPriceDiscount\_Cursor into @saleUnitPriceDiscount;

--Sale Line Total

declare SaleLineTotal\_Cursor cursor

for select d.LineTotal

from [SalesLT].[SalesOrderDetail] as d

left join [SalesLT].Product as p

on d.ProductID=p.ProductID

where d.SalesOrderID=@SalesOID;

open SaleLineTotal\_Cursor

fetch next from SaleLineTotal\_Cursor into @saleLineTotal;

print 'Product: ' + @productName + ' / OrderQty: ' + cast(@saleOrderQty as varchar(10)) + ' / UnitPrice: ' + cast(@saleUnitPrice as varchar(10)) + ' / UnitPriceDiscount: ' + cast(@saleUnitPriceDiscount as varchar(10)) + ' / Line Total: ' + cast(@saleLineTotal as varchar(10));

close ProductName\_Cursor;

close SaleOrderQty\_Cursor;

close SaleUnitPrice\_Cursor;

close SaleUnitPriceDiscount\_Cursor;

close SaleLineTotal\_Cursor;

deallocate ProductName\_Cursor;

deallocate SaleOrderQty\_Cursor;

deallocate SaleUnitPrice\_Cursor;

deallocate SaleUnitPriceDiscount\_Cursor;

deallocate SaleLineTotal\_Cursor;

set @salesCount = @salesCount - 1;

end

go

exec dbo.spListaCompra @SalesOID = 71863;

Etapa 2

--a)

create or alter function dbo.fnCodificaPassword (@password nvarchar(10))

returns nvarchar(10)

AS

begin

declare @newPassword nvarchar(10);

set @newPassword = hashbytes('SHA1',@password);

return @newPassword;

end;

select dbo.fnCodificaPassword('123abc.') as Password;

--b

CREATE TABLE CustomerPW (

ID int,

Password nvarchar(10)

);

--c

--d

Etapa 3

a)

b)

Etapa 4

a)

b)