



Quiz #1

Adrián Blanco Cordero

Universidad CENFOTEC

Programación de Bases de Datos

BRAYNER GERARDO SALMERON CASTILLO

Julio, 2024

## 1. Caso de Negocio

La empresa Northwind necesita una vista en SQL que facilite a los gerentes de ventas el acceso rápido a la información sobre las ventas de productos y su relación con los clientes. Los gerentes deben poder ver de manera rápida y clara los datos clave para evaluar el desempeño de las ventas y tomar decisiones informadas.

- **Requisitos de la Vista**

La vista debe llamarse SalesOverview y debe incluir los siguientes campos:

**CustomerID** - El ID del cliente.

**CompanyName** - El nombre de la compañía del cliente.

**OrderID** - El ID de la orden.

**OrderDate** - La fecha en que se realizó la orden.

**ProductID** - El ID del producto.

**ProductName** - El nombre del producto.

**Quantity** - La cantidad del producto ordenada.

**UnitPrice** - El precio unitario del producto.

**TotalPrice** - El precio total (calculado como  $\text{Quantity} * \text{UnitPrice}$ ).

### **Solucion:**

```
CREATE VIEW SalesOverview
as
SELECT C.CustomerID, C.CompanyName, O.OrderID, O.OrderDate, P.ProductID, P.ProductName,
OD.Quantity, OD.UnitPrice, (OD.Quantity * OD.UnitPrice) TotalPrice from Customers C
INNER JOIN Orders O on C.CustomerID = O.CustomerID
INNER JOIN [Order Details] OD on O.OrderID = OD.OrderID
INNER JOIN Products P on OD.ProductID = P.ProductID

select * from SalesOverview
```

## 2. Caso de Negocio

La empresa Northwind necesita un procedimiento almacenado que permita actualizar e insertar registros en la tabla Orders y la tabla Order Details. Este procedimiento debe utilizar transacciones para garantizar la integridad de los datos, manejar errores y devolver códigos de estado con descripciones específicas.

### Requisitos del Procedimiento Almacenado

**Transacciones:** Debe usar transacciones para asegurar la atomicidad de las operaciones.

**Manejo de Errores:** Debe manejar errores y devolver códigos de estado personalizados.

**Tablas Temporales:** Debe usar tablas temporales para actualizar e insertar registros.

**Códigos de Estado:**

- 200: Transacción exitosa.
- 400: Error controlado (p. ej., datos inválidos).
- 500: Error no controlado.

### Solucion:

```
CREATE OR ALTER PROCEDURE P_Insertar_Ordenes
```

```
    @OrderID int,  
    @CustomerID nchar(5),  
    @EmployeeID int,  
    @OrderDate datetime,  
    @RequiredDate datetime,  
    @ShippedDate datetime,  
    @ShipVia int,  
    @Freight money,  
    @ShipName nvarchar(40),  
    @ShipAddress nvarchar(60),  
    @ShipCity nvarchar(15),  
    @ShipRegion nvarchar(15),  
    @ShipPostalCode nvarchar(10),  
    @ShipCountry nvarchar(15),  
    @ProductID int,  
    @UnitPrice money,  
    @Quantity smallint,  
    @Discount real,  
    @StatusCode int OUTPUT,  
    @StatusMessage nvarchar(255) OUTPUT,
```

```

@TempOrderID int OUTPUT

AS
BEGIN

BEGIN TRANSACTION;

BEGIN TRY

CREATE TABLE #TempOrders(
                OrderID int,
CustomerID nchar(5),
EmployeeID int,
OrderDate datetime,
RequiredDate datetime,
ShippedDate datetime,
ShipVia int,
Freight money,
ShipName nvarchar(40),
ShipAddress nvarchar(60),
ShipCity nvarchar(15),
ShipRegion nvarchar(15),
ShipPostalCode nvarchar(10),
ShipCountry nvarchar(15)
);

CREATE TABLE #TempOrderDetails (
OrderID int,
ProductID int,
UnitPrice money,
Quantity smallint,
Discount real
);

INSERT INTO #TempOrders (OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate,
ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry)
VALUES (@OrderID, @CustomerID, @EmployeeID, @OrderDate, @RequiredDate, @ShippedDate,
@ShipVia, @Freight, @ShipName, @ShipAddress, @ShipCity, @ShipRegion, @ShipPostalCode,
@ShipCountry);

INSERT INTO #TempOrderDetails (OrderID, ProductID, UnitPrice, Quantity, Discount)
VALUES (@OrderID, @ProductID, @UnitPrice, @Quantity, @Discount);

IF EXISTS (SELECT 1 FROM Orders WHERE OrderID = @OrderID)
BEGIN
    SET @StatusCode = 400;
    SET @StatusMessage = 'Error en la entrada de datos. La orden ya existe.';
    SET @TempOrderID = NULL;
    RETURN;
END
ELSE
BEGIN

```

```

INSERT INTO Orders (CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia,
Freight, ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry)
SELECT CustomerID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight,
ShipName, ShipAddress, ShipCity, ShipRegion, ShipPostalCode, ShipCountry FROM #TempOrders;

SET @TempOrderID = SCOPE_IDENTITY();

INSERT INTO [Order Details](OrderID, ProductID, UnitPrice, Quantity, Discount)
SELECT @TempOrderID, ProductID, UnitPrice, Quantity, Discount FROM #TempOrderDetails;
END

COMMIT TRANSACTION;

SET @StatusCode = 200;
SET @StatusMessage = 'Transacción exitosa.';

END TRY
BEGIN CATCH

IF @@TRANCOUNT > 0
BEGIN
ROLLBACK TRANSACTION;
END

DECLARE @ErrorMessage nvarchar(4000);
SELECT @ErrorMessage = ERROR_MESSAGE();

SET @StatusCode = 500;
SET @StatusMessage = 'Error no controlado: ' + @ErrorMessage;

END CATCH
END;
GO

```

**3. (Opcional) Por 5 puntos extra, explique cómo se ha sentido en el curso, tanto con el docente como con el contenido del curso.**

El curso me ha parecido bastante interesante, me ha sacado mucho de mi zona de confort que es la programación de aplicaciones y programación web, me ha tocado estudiar bastante, todavía no me siento al 100 con la materia y necesito seguir estudiando, me ha tocado ver bastantes videos en youtube sobre los temas para entenderlos mas a fondo, pero a pesar de eso siento que voy saliendo bien con las cosas.