

TRIGGERS

--1.- Utilizando trigger, validar que solo se vendan ordenes de lunes a viernes.

```
create trigger tr_ejercicio1
```

```
on Orders
```

```
for insert as
```

```
declare @dia int
```

```
select @dia = DATEPART(dw, OrderDate) from inserted
```

```
begin
```

```
    if @dia in (1,7)
```

```
        rollback tran
```

```
        raiserror('Error, no puedes ingresar ventas los dias sabado y
```

```
domingo',16,1)
```

```
end
```

```
go
```

--2.- Validar que no se vendan mas de 20 ordenes por empleado en una semana.

```
create or alter trigger tr_ejercicio2
```

```
on Orders
```

```
for insert as
```

```
declare @Ordenes int, @EmpId int, @Fecha datetime
```

```
select @EmpId = Employeeid from inserted
```

```
select @Fecha = OrderDate from inserted
```

```
select @Ordenes = Count(OrderId) from Orders where EmployeeID = @EmpId and
```

```
datepart(ww, OrderDate) = datepart(ww,@Fecha)
```

```
if @Ordenes > 20
```

```
begin
```

```
    rollback tran
```

```
    raiserror('Error, no puedes ingresar mas de 20 ordenes por empleado en una
```

```
misma semana',16,1)
```

```
end
```

```
go
```

--3.- Validar que el campo firstname en la tabla employees solamente tenga nombres que inicien con vocal.

```
create or alter trigger tr_ejercicio3
```

```
on Employees
```

```
for insert as
```

```
declare @firstname nvarchar(30)
```

```
select @firstname = FirstName from inserted
```

```
if @firstname like '[AEIOU]%'
```

```
begin
```

```
    rollback tran
```

```
    raiserror('Error, solo puedes ingresar nombres que empiecen con vocal',16,1)
```

```
end
```

```
go
```

```
select FirstName from Employees where FirstName not like '[AEIOU]' and FirstName
```

```
not like '[AEIOU]%'
```

```
go
```

--4.- validar que el importe de venta de cada orden no sea mayor a \$10,000.

```
create or alter trigger tr_ejercicio4
```

```
on [Order Details]
```

```
for insert as
```

```
declare @importe int
```

```
select @importe = Quantity * UnitPrice from inserted
```

```
if @importe > 10000
```

```

begin
    rollback tran
    raiserror('No puedes ingresar un importe mayor a 10000',16,1)
end
go

--5.- validar que no se puedan eliminar ordenes que se hicieron los lunes.
create or alter trigger tr_ejercicio5
on Orders
for delete as
declare @dia int
select @dia = DatePart(dw, OrderDate) from deleted
if @dia = 2
begin
    rollback tran
    raiserror('No puedes borrar registros de ordenes realizadas en lunes',16,1)
end
go

--6.- Validar que no se realicen inserciones masivas en la tabla products.
create or alter trigger tr_ejercicio6
on Products
for insert as
declare @count int
select @count = Count(*) from inserted
if (@count > 1)
begin
    rollback tran
    raiserror('No se permiten inserciones masivas',16,1)
end
go

--7.- Validar que no se pueda modificar el campo unitprice de la tabla [order
details].
create trigger tr_ejercicio7
on [Order Details]
for update as

if update(unitprice)
begin
    rollback tran
    raiserror('Error, no puedes modificar este campo',16,1)
end

--8.- Validar que solo se pueda actualizar una sola vez el nombre del cliente.
Alter Table Customers add nombreAct int
update Customers set nombreAct = 0
go

create or alter trigger tr_ejercicio8
on Customers
for update as
declare @clave int, @conta int
select @clave = CustomerID, @conta = isnull(nombreAct, 0) from inserted
if update(ContactName)
begin
    if @conta > 1
    begin

```

```

        rollback tran
        raiserror('Error, no se puede actualizar',16,1)
    end
    else
    begin
        update Customers set nombreAct = @conta + 1 where CustomerID = @clave
    end
end
go

--9.- Validar que no se puedan eliminar categorías que tengan una clave impar.
create or alter trigger tr_ejercicio9
on Categories
for delete as
declare @clave int
select @clave = CategoryId from deleted
begin
    if (@clave % 2 != 0)
    begin
        rollback tran
        raiserror ('Error, no puedes borrar claves impares',16,1)
    end
end
go

--10.- Validar que no se puedan insertar ordenes que se realicen en domingo.
create or alter trigger tr_ejercicio10
on Orders
for insert as
declare @dia int
select @dia = datepart(dw, OrderDate) from inserted
if @dia = 1
begin
    rollback tran
    raiserror('Error, no puedes insertar ordenes en domingo',16,1)
end
go

```

CONSULTAS

```
USE Northwind
```

```
/* Consulta con todos los datos de todos los empleados */  
SELECT * FROM Employees
```

```
/* Consulta con el nombre del cliente y su dirección */  
SELECT CompanyName, ADDRESS FROM Customers
```

```
/* Consulta con clave y el nombre del empleado con encabezado */  
SELECT Clave = EmployeeID, FirstName AS Nombre FROM Employees
```

```
SELECT 'Clave del Empleado' = EmployeeID, FirstName AS [Nombre del Empleado] FROM  
Employees
```

```
/* Concatenar texto a un campo de texto */
```

```
/* Muestra el precio de los productos aumentando 10%, 20% y 30% */  
SELECT ProductName,  
'Original Price' = UnitPrice,  
'Aumento de 10%' = UnitPrice * 1.1,  
'Aumento de 20%' = UnitPrice * 1.2,  
'Aumento de 30%' = UnitPrice * 1.3  
FROM Products
```

```
/* Imprimir el aumento simulado de 20 pesos a todos los productos */  
SELECT ProductName, UnitPrice, 'Aumento de 20 Pesos' = UnitPrice + 20  
FROM Products
```

```
/* Tres elevado a la cuarta potencia */  
SELECT ABS(-21)  
SELECT FLOOR(3.9)  
SELECT CEILING(3.1)
```

```
SELECT ROUND(12.23456, 2)  
SELECT SIGN(12.3)  
SELECT SIGN(-12.3)  
SELECT RAND()
```

```
SELECT POWER(3,4)
```

```
/* Elevar al cuadrado el precio de los productos */  
SELECT ProductName, POWER(UnitPrice, 2) FROM Products
```

```
/* Imprimir la raíz cuadrada del precio de los productos */  
SELECT ProductName, UnitPrice, 'Raiz Cuadrada' = SQRT(UNITPRICE)  
FROM Products
```

```
/* */  
SELECT SUBSTRING('Culiacan, Sinaloa', 1, 5)  
SELECT RIGHT('Culiacan, Sinaloa', 7)  
  
SELECT ASCII('A')
```

```
SELECT ASCII('a')
```

```
SELECT CHAR(5)
```

```
SELECT 'El precio es: ' + 45.34
```

```
SELECT 'El precio es: ' + STR(45.34, 50, 2)
```

```
SELECT 'El precio es: ' + CONVERT(CHAR(30), 45.34)
```

```
SELECT 'El precio es: ' + CAST(45.34 AS CHAR(30))
```

```
SELECT 'Ana' + SPACE(100) + 'Lara'
```

```
SELECT DIFFERENCE('Lara', 'Lara')
```

```
SELECT DIFFERENCE('Lara', 'Jorge')
```

```
SELECT UPPER('lara')
```

```
SELECT LOWER('CASTRO')
```

```
/* Mostrar el nombre completo del empleado */
```

```
SELECT FirstName + CHAR(32) + ' ' + SPACE(1) + LastName FROM Employees
```

```
/* Mostrar el nombre del empleado como J. Perez */
```

```
SELECT SUBSTRING(FirstName, 1, 1) + '.' + SPACE(1) + LastName FROM Employees
```

```
/* Mostrar en mayusculas el nombre completo de empleado */
```

```
SELECT UPPER(FirstName + ' ' + LastName) FROM Employees
```

```
/* Mostrar la ultima letra del apellido del empleado */
```

```
SELECT LastName,  
       Apellido = RIGHT(LastName, 1),  
       Apellido = SUBSTRING(LastName, LEN(LastName), 1)  
FROM Employees
```

```
/* Consulta de empleados: Jose Lopez nacio en el dia Jueves 8 de enero de 1970 */
```

```
SELECT FirstName, LastName, BirthDate FROM Employees
```

```
SELECT FirstName + ' ' + LastName + ' nació el día ' +  
       DATENAME(dw, BirthDate) + ' ' +  
       DATENAME(dd, BirthDate) + ' de ' +  
       DATENAME(mm, BirthDate) + ' de ' +  
       CONVERT(CHAR(4), DATEPART(yy, BirthDate))  
FROM Employees
```

```
/* Función que regresa la fecha del servidor */
```

```
SELECT GETDATE()
```

```
/* Consulta con los años vividos por los empleados (edad) */
```

```
SELECT EmployeeID, FirstName, BirthDate,  
       DATEDIFF(yy, BirthDate, GETDATE()),  
       YEAR(GETDATE()) - YEAR(BirthDate)  
FROM Employees
```

```
/* Consulta con el nombre y la antigüedad de los empleados */
```

```

SELECT FirstName, HireDate, Antigüedad = DATEDIFF(yy, HireDate, GETDATE())
FROM Employees

/* Consulta con la edad del empleado cuando entró a trabajar */
SELECT FirstName, BirthDate, HireDate, 'Edad cuando entro a trabajar' = DATEDIFF(yy,
BirthDate, HireDate)
FROM Employees

/* Consulta con los productos con precio menor a 30 */
SELECT * FROM Employees
WHERE YEAR(BirthDate) < 1960

/* Consulta con los empleados que nacieron antes del 1960 */
SELECT * FROM Employees
WHERE YEAR(BirthDate) < 1960

/* Consulta con los productos con un precio entre 20 y 50 */
SELECT * FROM Products WHERE UnitPrice BETWEEN 20 AND 50

/* Consulta con las ordenes del primer semestre de 1998 */
SELECT * FROM Orders WHERE OrderDate Between '1-1-1998' AND '6-30-1998'

/* Productos que valgan 10, 20 o 31 */
SELECT * FROM Products WHERE UnitPrice IN (10, 20, 31)

/* Consulta con los productos donde su nombre sea ikura */
SELECT * FROM Products WHERE ProductName LIKE 'ikura'

/* Consulta con los productos que empiecen con el texto "Queso" */
SELECT * FROM Products WHERE ProductName LIKE 'Queso%'

/* Consulta con los productos que terminen con la cadena "es" */
SELECT * FROM Products WHERE ProductName LIKE '%es'

/* Consulta con los productos que contengan la cadena "as" */
SELECT * FROM Products WHERE ProductName LIKE '%as%'

/* Consulta con los productos que empiecen con la letra G y terminen con la letra A
*/
SELECT * FROM Products WHERE ProductName LIKE 'g%a'

/* Consulta con los productos que empiecen con la letra M, G, R */
SELECT * FROM Products WHERE Productname LIKE '[mgr]%'

SELECT * FROM Products
WHERE ProductName LIKE 'm%' OR ProductName LIKE 'g%' OR ProductName LIKE 'r%'

SELECT * FROM Products
WHERE SUBSTRING(ProductName, 1, 1) IN ('m', 'g', 'r')

/* Consulta con los productos que terminen con consonantes */
SELECT * FROM Products WHERE ProductName LIKE '%[^aeiou]'

```

```
SELECT * FROM Products WHERE ProductName NOT LIKE '%[aeiou]'
```

```
/* Consulta con los productos que tengan 5 caracteres */
```

```
SELECT * FROM Products WHERE ProductName LIKE '_____'
```

```
SELECT * FROM Products WHERE LEN(ProductName) = 5
```

```
/* Consulta con los productos que en la tercera posicion tengan una VOCAL */
```

```
SELECT * FROM Products WHERE Productname LIKE '__[aeiou]%'
```

```
/* Productos que su primera palabra tenga 5 caracteres */
```

```
SELECT * FROM Products WHERE ProductName LIKE '_____%'
```

```
/* Consulta con los empleados que no tienen asignada una region */
```

```
SELECT * FROM Employees WHERE Region IS NULL
```

```
-- Est es un error
```

```
SELECT * FROM Employees WHERE Region = NULL
```

```
/* Consulta con los clientes que si tienen asignado un Fax */
```

```
SELECT CustomerID, CompanyName, Fax FROM Customers  
WHERE Fax IS NOT NULL
```

```
/* consulta con los nombre de los empleados ordenados por apellido */
```

```
SELECT EmployeeID, LastName, FirstName FROM Employees  
ORDER BY LastName ASC
```

```
/* Consulta con los productos ordenados de mayor a menor precio */
```

```
SELECT ProductID, ProductName, UnitPrice FROM Products  
ORDER BY UnitPrice DESC
```

```
SELECT ProductID, ProductName, UnitPrice FROM Products  
ORDER BY 3 DESC
```

```
/* Consulta con los 5 productos mas caros */
```

```
SELECT TOP 5 ProductID, ProductName, UnitPrice FROM Products  
ORDER BY UnitPrice DESC
```

```
/* Consulta con los 2 empleados mas jóvenes */
```

```
SELECT TOP 2 EmployeeID, FirstName, BirthDate FROM Employees  
ORDER BY BirthDate DESC
```

```
/* Consulta con las ultmas 5 ordenes de 1996 del Empleado 2 */
```

```
SELECT TOP 5 OrderID, OrderDate, EmployeeID  
FROM Orders  
WHERE EmployeeID = 2 AND YEAR(OrderDate) = 1996  
ORDER BY OrderDate DESC
```

```
/* Consulta con los 2 productos mas baratos del proveedor 2 */
```

```

SELECT TOP 2 ProductID, ProductName, UnitPrice, SupplierID
FROM Products
WHERE SupplierID = 2
ORDER BY UnitPrice ASC

```

```

/* Cross Join, Combinaciones cruzadas */
-- 10 Columnas, 77 Renglones
SELECT * FROM Products

```

```

-- 4 Columnas, 8 Renglones
SELECT * FROM Categories

```

```

SELECT * FROM Products CROSS JOIN Categories
SELECT * FROM Products, Categories

```

```

-- Columnas: 10 + 4 = 14
-- Renglones: 77 * 8 = 616

```

```

-- Columnas 12, Renglones 29
SELECT * FROM Suppliers

```

```

SELECT * FROM Products CROSS JOIN Categories CROSS JOIN Suppliers

```

```

--Columnas: 10 + 4 + 2 = 26
-- Renglones: 77 * 8 * 29 = 17,864

```

```

/* Consulta con el nombre del producto y nombre de la categoria */
-- ANSI
SELECT Products.ProductName, Categories.CategoryName
FROM Products
INNER JOIN Categories ON Categories.CategoryID = Products.CategoryID

```

```

SELECT P.ProductName, C.CategoryName
FROM Products P
INNER JOIN Categories C ON C.CategoryID = P.CategoryID

```

```

-- Con transact-SQL
SELECT p.ProductName, C.CategoryName
FROM Products p, categories c
WHERE c.CategoryID = p.CategoryID

```

```

/*      COnsulta con los productos con un precio menores a 20 pesos,
      Mostrar el nombre del producto, precio, nombre de la categoria */
SELECT p.Productname, p.UnitPrice, p.CategoryID, c.CategoryID, c.CategoryName
FROM Products p
INNER JOIN Categories c ON p.CategoryID = c.CategoryID
WHERE p.UnitPrice < 20

```

```

/*      Consulta con la clave y fecha de la orden, nombre del empleado y nombre del
cliente,

```



```

mostrar solamente las ordenes realizadas en 1996 */
SELECT o.OrderID, o.OrderDate, e.FirstName + ' ' + e.LastName, c.CompanyName
FROM Orders o
INNER JOIN Employees e ON e.EmployeeID = o.EmployeeID
INNER JOIN Customers c ON c.CustomerID = o.CustomerID
WHERE
YEAR(o.OrderDate) = 1996

/* Consulta con la clave de la orden, nombre del producto, cantidad, preio y
total de la venta.
Mostrar solo las ordenes realizadas los dias lunes */
SELECT o.OrderID, Day = DATENAME(dw, o.OrderDate), p.ProductName, d.Quantity,
d.UnitPrice,
Total = d.Quantity * d.UnitPrice
FROM Orders o
INNER JOIN [Order Details] d ON d.OrderId = o.OrderID
INNER JOIN Products p ON p.ProductID = d.ProductID
WHERE
DATEPART(dw, o.OrderDate) = 2

/* Consulta con el nombre del empleado y nombre del territorio */
SELECT Employee = e.Firstname + ' ' + e.LastName, t.TerritoryDescription
FROM Employees e
INNER JOIN EmployeeTerritories et ON et.EmployeeID = e.EmployeeID
INNER JOIN Territories t ON t.TerritoryID = et.TerritoryID

SELECT * FROM EmployeeTerritories

/* Consulta con el nombre del empleado y nombre de su jefe */
SELECT ClaveEmp = e.EmployeeID, Empleado = e.FirstName + ' ' + e.LastName,
ClaveJefe = j.EmployeeID, Jefe = j.FirstName + ' ' + j.LastName
FROM Employees e
INNER JOIN Employees J ON e.ReportsTo = j.EmployeeID

SELECT EmployeeID, ReportsTo FROM Employees

```