

# Ejercicios 4 de Programación en Transact-SQL - Soluciones

USE Gestion8

```
-- 1.- Ahora vamos a refinar el procedimiento para que el valor devuelto nos permita
saber si el representante existe o no.
-- Fijaremos que el procedimiento devuelve un valor imposible (por ejemplo -1) si el
representante no existe.
IF object_id('CuantosPedRep3_3', 'P') IS NOT NULL DROP PROC CuantosPedRep3_3;
GO
CREATE PROCEDURE CuantosPedRep3_3 @rep INT
AS
    IF NOT EXISTS(SELECT * FROM empleados WHERE numemp=@rep)
        RETURN -1
    ELSE
        RETURN (SELECT count(*) FROM pedidos WHERE rep=@rep);
GO
PRINT 'Prueba del ejercicio 1'
DECLARE @resul INT
EXEC @resul = CuantosPedRep3_3 101 -- Un representante que tiene varios pedidos
IF @resul = -1 PRINT 'El representante 101 no existe'
ELSE
    IF @resul = 0 PRINT 'El representante 101 no tiene pedidos'
    ELSE PRINT 'El representante 101 tiene ' + STR(@resul,3) + ' pedidos'
EXEC @resul = CuantosPedRep3_3 104 -- Un representante que no tiene pedidos
IF @resul = -1 PRINT 'El representante 104 no existe'
ELSE
    IF @resul = 0 PRINT 'El representante 104 no tiene pedidos'
    ELSE PRINT 'El representante 104 tiene ' + STR(@resul,3) + ' pedidos'
EXEC @resul = CuantosPedRep3_3 200 -- Un representante que no existe
IF @resul = -1 PRINT 'El representante 200 no existe'
ELSE
    IF @resul = 0 PRINT 'El representante 200 no tiene pedidos'
    ELSE PRINT 'El representante 200 tiene ' + STR(@resul,3) + ' pedidos'

-- 2.- Ahora vamos a refinarlo más, es muy engorroso repetir por cada ejecución el IF
con los diferentes PRINT según el caso.
-- Así que vamos a definir un procedimiento para solucionar el problema vamos a
definir un nuevo procedimiento que nos imprima el resultado.
-- Es decir vamos a crear un nuevo procedimiento que imprima el nº de pedidos de un
determinado representante pero ese procedimiento utilizará el procedimiento anterior
para saber cuántos pedidos tiene el representante.
IF object_id('ImprimePedRep', 'P') IS NOT NULL DROP PROC ImprimePedRep;
GO
CREATE PROCEDURE ImprimePedRep @rep INT
AS
    DECLARE @resul INT
    EXEC @resul = CuantosPedRep3_3 @rep
    IF @resul = -1 PRINT 'El representante ' + STR(@rep,3) + ' no existe'
    ELSE
        IF @resul = 0 PRINT 'El representante ' + STR(@rep,3) + ' no tiene
pedidos'
        ELSE PRINT 'El representante ' + STR(@rep,3) + ' tiene ' + STR(@resul,3)
+ ' pedidos'
GO
PRINT 'Prueba del ejercicio 2'
EXEC ImprimePedRep 101 -- Un representante que tiene varios pedidos
EXEC ImprimePedRep 104 -- Un representante que no tiene pedidos
EXEC ImprimePedRep 200 -- Un representante que no existe

-- 3.- Redactar un procedimiento que nos devuelva el precio de un determinado
artículo.
-- Si el producto no existe el procedimiento devolverá -1, si no tiene precio
devolverá 0.
```

```

-- Aquí estamos obligados a utilizar un parámetro de salida porque el precio no es un
entero.
-- Ojo, cuando se concatena una cadena con un valor NULL el resultado es NULL.
IF object_id('PrecioArti4_1', 'P') IS NOT NULL DROP PROC PrecioArti4_1;
GO
CREATE PROCEDURE PrecioArti4_1 @fab CHAR(20), @prod CHAR(20), @precio MONEY OUTPUT
AS
    SET @precio = (SELECT ISNULL(precio,0) FROM productos WHERE idfab=@fab AND
idproducto=@prod);
    -- Si el artículo existe pero no tiene precio, devolverá 0
    SET @precio = ISNULL(@precio, -1) -- Si @precio es nulo es porque la SELECT no
ha devuelto ningún valor (el producto no existe). Entonces sustituimos el valor NULL
por un -1
GO
PRINT 'Prueba del ejercicio 3'
DECLARE @precio MONEY
EXEC PrecioArti4_1 'aci', '41001', @Precio OUTPUT -- Un producto que no tiene precio
PRINT 'El precio es ' + STR(@Precio,8,2)
EXEC PrecioArti4_1 'bic', '41003', @Precio OUTPUT -- Un producto que tiene precio
PRINT 'El precio es ' + STR(@Precio,8,2)
EXEC PrecioArti4_1 'aci', '50000', @Precio OUTPUT -- Un producto que no existe
PRINT 'El precio es ' + STR(@Precio,8,2)
-- La visualización de los resultados se puede mejorar como hemos hecho en ejercicios
anteriores.
-- Si quieres practicar más puedes hacerlo.

-- 4.- Redacta un procedimiento que nos devuelva la descripción de un determinado
artículo.
-- En este caso en la tabla productos el campo descripcion no admite valores nulos
por lo que sólo nos tenemos que preocupar de si el artículo existe o no.
IF object_id('DescArti4_1', 'P') IS NOT NULL DROP PROC DescArti4_1;
GO
CREATE PROCEDURE DescArti4_1 @fab CHAR(20), @prod CHAR(20), @descripcion CHAR(20)
OUTPUT
AS
    SET @descripcion = (SELECT descripcion FROM productos WHERE idfab=@fab AND
idproducto=@prod);
    SET @descripcion = ISNULL(@descripcion, 'Producto no existe')
GO
PRINT 'Prueba del ejercicio 4'
DECLARE @desc CHAR(20)
EXEC DescArti4_1 'aci', '41001', @desc OUTPUT -- Un producto que existe
PRINT 'La descripción es ' + @desc
EXEC DescArti4_1 'aci', '50000', @desc OUTPUT -- Un producto que no existe
PRINT 'La descripción es ' + @desc

-- 5.- ¿Qué habría que cambiar del procedimiento anterior si la descripción admitiese
valores nulos?
-- Si la columna descripción admitiese valores nulos, los productos con descripción
nula saldrían como que no existen. lo que haremos será devolver en la descripción un
texto como No tiene descripción en el caso de que el producto no tenga descripción.
-- Para poder probar el procedimiento tenemos que cambiar la definición de la tabla
productos para permitir valores nulos en descripción.
-- También tenemos que añadir un nuevo producto y no asignarle descripción.

IF object_id('DescArti4_2', 'P') IS NOT NULL DROP PROC DescArti4_2;
GO
CREATE PROCEDURE DescArti4_2 @fab CHAR(20), @prod CHAR(20), @descripcion CHAR(20)
OUTPUT
AS
    SET @descripcion = (SELECT ISNULL(descripcion,'No tiene descripción') FROM
productos WHERE idfab=@fab AND idproducto=@prod);
    SET @descripcion = ISNULL(@descripcion, 'Producto no existe')
GO
PRINT 'Prueba del ejercicio 5'
DECLARE @desc CHAR(20)

```

```

EXEC DescArti4_2 'aci', '41001', @desc OUTPUT -- Un producto con descripción
PRINT 'La descripción es ' + @desc
EXEC DescArti4_2 'rei', '1111', @desc OUTPUT -- Un producto que no tiene descripción
(el que hemos creado nuevo para la prueba)
PRINT 'La descripción es ' + @desc
EXEC DescArti4_2 'aci', '50000', @desc OUTPUT -- Un producto que no existe
PRINT 'La descripción es ' + @desc
--
-- 6.- El cliente nos pide que si el producto no tiene descripción que no aparezca
nada en la descripción, y si el producto no existe el procedimiento debe devolver -1.

IF object_id('DescArti4_3', 'P') IS NOT NULL DROP PROC DescArti4_3;
GO
CREATE PROCEDURE DescArti4_3 @fab CHAR(20), @prod CHAR(20), @descripcion CHAR(20)
OUTPUT
AS
    IF NOT EXISTS (SELECT * FROM Productos WHERE idfab=@fab AND idproducto=@prod)
        RETURN (-1)
    ELSE
        SET @descripcion = (SELECT descripcion FROM productos WHERE idfab=@fab
AND idproducto=@prod);
GO
PRINT 'Prueba del ejercicio 6'
DECLARE @desc CHAR(20), @resul INT
EXEC DescArti4_3 'aci', '41001', @desc OUTPUT -- Un producto con descripción
PRINT 'La descripción es ' + ISNULL(@desc, ' ') -- Recuerda que si concatenas con NULL
el resultado da NULL
EXEC DescArti4_3 'rei', '1111', @desc OUTPUT -- Un producto que no tiene descripción
(el que hemos creado nuevo para la prueba)
PRINT 'La descripción es ' + ISNULL(@desc, ' ')
EXEC @resul=DescArti4_3 'aci', '50000', @desc OUTPUT -- Un producto que no existe
IF @resul = -1
    PRINT 'El producto no existe'
ELSE
    PRINT 'La descripción es ' + ISNULL(@desc, ' ')

-- 7.- Redacta un procedimiento que nos devuelva el precio medio de venta de un
determinado artículo un determinado mes del año en curso.
-- Aquí puede ocurrir que tengamos pedidos del artículo, que tengamos pedidos del
artículo pero no en el periodo indicado, que no tengamos pedidos del artículo, y que
el artículo no exista.
-- Pero consideraremos tres casos: que tenga pedidos en el periodo indicado, en este
caso el resultado será el precio medio del artículo
-- que el artículo no tenga pedidos (ninguno o ninguno en el periodo) en este caso se
devolverá precio cero.
-- que el artículo no exista en este caso se devolverá -1.
IF object_id('PrecioMensual4_1', 'P') IS NOT NULL DROP PROC PrecioMensual4_1;
GO
CREATE PROCEDURE PrecioMensual4_1 @fab CHAR(20), @prod CHAR(20), @mes INT, @precio
MONEY OUTPUT
AS
    SET @precio = (SELECT ROUND(AVG(importe/cant),2) FROM pedidos
WHERE fab=@fab AND producto=@prod AND MONTH(fechapedido) = @mes
AND YEAR(fechapedido) = YEAR(GETDATE()));
    -- Si hay pedidos del artículo en el mes indicado la media no puede ser
nula porque los campos no admiten nulos
    -- redondeamos a dos decimales
-- Ahora el precio puede ser nulo por dos razones, porque no hayan pedidos o porque
no exista el artículo
    IF NOT EXISTS (SELECT * FROM productos WHERE idfab=@fab AND idproducto=@prod)
        SET @precio = -1
    ELSE
        SET @precio = ISNULL(@precio, 0) -- Si @precio es nulo y estamos
en el ELSE (el artículo existe) es porque la SELECT
-- no ha devuelto ningún valor (el producto no tiene pedidos).
GO

```

```

PRINT 'Prueba del ejercicio 7'
DECLARE @precio MONEY
EXEC PrecioMensual4_1 'aci', '41004', 5, @Precio OUTPUT -- Un producto que tiene
pedidos en el mes indicado del año actual (si hace falta cambia los pedidos)
PRINT 'El precio es ' + STR(@Precio,8,2)
EXEC PrecioMensual4_1 'aci', '41004', 7, @Precio OUTPUT -- Un producto que tiene
pedidos en el mes indicado del año actual (si hace falta cambia los pedidos)
PRINT 'El precio es ' + STR(@Precio,8,2)
EXEC PrecioMensual4_1 'aci', '11111', 5, @Precio OUTPUT -- Un producto que no existe
PRINT 'El precio es ' + STR(@Precio,8,2)
-- Repetir la prueba para un producto que tiene pedidos en el mes indicado pero de
otro año
-- Para un producto que tiene pedidos del año actual pero no en el mes indicado
-- Para un producto que tiene pedidos pero no del año actual pero ni en el mes
indicado
-- Para un producto que no tiene pedidos
-- Para un producto que no existe
-- * Como en la base de datos no existen pedidos para todos los casos podéis añadir
más filas para las pruebas.

-- 8.- Redactar un procedimiento que nos devuelva el código de la oficina que tiene
más empleados, si hay más de una, devolverá el código mayor.
IF object_id('OficinaMayor4_1', 'P') IS NOT NULL DROP PROC OficinaMayor4_1;
GO
CREATE PROCEDURE OficinaMayor4_1
AS
    RETURN (SELECT TOP 1 oficina FROM empleados WHERE oficina IS NOT NULL GROUP BY
oficina ORDER BY COUNT(*) DESC, oficina DESC);
GO
PRINT 'Prueba del ejercicio 8'
DECLARE @ofi INT
EXEC @ofi = OficinaMayor4_1
PRINT 'La oficina con mayor número de empleados es la: ' + STR(@ofi,3)

--9. Crea un procedimiento de nos devuelva cuántas oficinas se encuentran en una
determinada región. Házlo con un parámetro de salida.
IF object_id('OficinasReg4_1', 'P') IS NOT NULL DROP PROC OficinasReg4_1;
GO
CREATE PROC OficinasReg4_1 @reg CHAR(20) , @cuantas INT OUTPUT
AS
    SET @cuantas=(SELECT COUNT(*) FROM oficinas WHERE region=@reg);
GO
-- Comprobación
PRINT 'Prueba del ejercicio 9'
DECLARE @resul INT
EXEC OficinasReg4_1 'este', @resul OUTPUT ; -- Una region con más de 5 oficinas
PRINT @resul
EXEC OficinasReg4_1 'nada', @resul OUTPUT; -- Una región que no tiene oficinas
PRINT @resul

--10. Crea un procedimiento de nos devuelva cuántas oficinas se encuentran en una
determinada región. Házlo con RETURN.
IF object_id('OficinasReg4_2', 'P') IS NOT NULL DROP PROC OficinasReg4_2;
GO
CREATE PROC OficinasReg4_2 @reg CHAR(20)
AS
    RETURN(SELECT COUNT(*) FROM oficinas WHERE region=@reg);
GO
-- Comprobación
PRINT 'Prueba del ejercicio 10'
DECLARE @resul INT
EXEC @resul = OficinasReg4_2 'este'; -- Una region con oficinas
PRINT @resul
EXEC @resul = OficinasReg4_2 'nada'; -- Una región que no tiene oficinas
PRINT @resul

```