

Ejercicios 2 de Programación en Transact-SQL - Soluciones

```
USE Gestion8;
GO
Como vamos a definir todos los procedimientos en Gestion8 escribimos el USE y después
el GO porque después viene un CREATE PROCEDURE.

-- 1.- Listar los datos de los empleados de una determinada oficina así como los
datos de la oficina, no sacar ningún mensaje adicional. Si la oficina no tiene
empleados se sacarán sus datos y nulos en los campos de los empleados.
(EmpOficina2_1)

-- El procedimiento tiene un parámetro de entrada el nº de la oficina, no es
opcional, y necesitamos sacar la información de las dos tablas.

IF object_id('EmpOficina2_1', 'P') IS NOT NULL DROP PROC EmpOficina2_1;
GO
CREATE PROCEDURE EmpOficina2_1 @ofi INT
AS
    SELECT * FROM oficinas O LEFT JOIN empleados E ON O.oficina=E.oficina
    WHERE O.oficina=@ofi;
GO
EXEC EmpOficina2_1      -- Para comprobar que NO funciona sin parámetros
EXEC EmpOficina2_1 21  -- Una oficina que tiene varios empleados
EXEC EmpOficina2_1 24  -- Una oficina que no tiene empleados
EXEC EmpOficina2_1 300 -- Una oficina que no existe

-- 2.- Para que la salida obtenida quede más clara, delante de la lista deberá
aparecer el texto "Detalle de la oficina xx:" Donde xx será el nº de la oficina.

-- Si queremos que aparezca un texto y en la pestaña resultados (como la lista) hay
que utilizar un SELECT mensaje.

IF object_id('EmpOficina2_2', 'P') IS NOT NULL DROP PROC EmpOficina2_2;
GO
CREATE PROCEDURE EmpOficina2_2 @ofi INT
AS
    SELECT 'Detalle de la oficina ' + STR(@ofi,3) + ': ' AS [ ]
    SELECT * FROM oficinas O LEFT JOIN empleados E ON O.oficina=E.oficina
    WHERE O.oficina=@ofi;
GO
EXEC EmpOficina2_2      -- Para comprobar que NO funciona sin parámetros
EXEC EmpOficina2_2 21  -- Una oficina que tiene varios empleados
EXEC EmpOficina2_2 24  -- Una oficina que no tiene empleados
EXEC EmpOficina2_2 300 -- Una oficina que no existe

-- 3.- Y si la oficina no existe, después del listado saca un texto que diga "La
oficina xx no existe". Donde xx será el nº de la oficina.

-- Tenemos que añadir otro SELECT con el mensaje después de la SELECT FROM Oficinas,
pero el mensaje no tiene que aparecer siempre, sólo en el caso de que no exista la
oficina, luego lo tenemos que poner dentro de un IF. ¿Cómo preguntaremos si la
oficina existe?
Podemos utilizar la función NOT EXISTS que ya hemos visto en algún ejercicio:
IF NOT EXISTS (SELECT * FROM oficinas WHERE oficina=@ofi)
Pero también podemos utilizar la función del sistema @@ROWCOUNT que nos devuelve el
nº de filas afectadas por la última operación. De esta forma evitamos hacer una
SELECT adicional y el resultado es más rápido.

IF object_id('EmpOficina2_3', 'P') IS NOT NULL DROP PROC EmpOficina2_3;
GO
CREATE PROCEDURE EmpOficina2_3 @ofi INT
AS
    SELECT 'Detalle de la oficina ' + STR(@ofi,3) + ': ' AS [ ]
    SELECT * FROM oficinas O LEFT JOIN empleados E ON O.oficina=E.oficina
```

```

WHERE O.oficina=@ofi;
IF @@ROWCOUNT = 0 SELECT 'La oficina ' + str(@ofi,3) + ' no existe' AS [ ]
GO
EXEC EmpOficina2_3      -- Para comprobar que NO funciona sin parámetros
EXEC EmpOficina2_3 21  -- Una oficina que tiene varios empleados
EXEC EmpOficina2_3 24  -- Una oficina que no tiene empleados
EXEC EmpOficina2_3 300 -- Una oficina que no existe

-- 4.- Ahora, si la oficina oficina no existe tiene que aparecer el mensaje pero no
la tabla vacía.

-- para que la lista (el resultado de la SELECT) sólo aparezca sólo en algunos casos,
hay que ponerla dentro de un IF. ¿Cuál es la condición para que salga la lista? Que
la oficina exista, luego hay que poner SI (la oficina existe) entonces SELECT...

-- Y el IF que teníamos para saber cuándo enviar el mensaje de oficina no existe ya
no hace falta, sólo tenemos que poner el mensaje dentro del ELSE.
IF object_id('EmpOficina2_4', 'P') IS NOT NULL DROP PROC EmpOficina2_4;
GO
CREATE PROCEDURE EmpOficina2_4 @ofi INT
AS
    SELECT 'Detalle de la oficina ' + STR(@ofi,3) + ': ' AS [ ]
    IF EXISTS(SELECT * FROM oficinas WHERE oficina=@ofi)
        SELECT * FROM oficinas O LEFT JOIN empleados E ON O.oficina=E.oficina
        WHERE O.oficina=@ofi;
    ELSE SELECT 'La oficina ' + STR(@ofi,3) + ' no existe' AS [ ]
GO
EXEC EmpOficina2_4      -- Para comprobar que NO funciona sin parámetros
EXEC EmpOficina2_4 21  -- Una oficina que tiene varios empleados
EXEC EmpOficina2_4 24  -- Una oficina que no tiene empleados
EXEC EmpOficina2_4 300 -- Una oficina que no existe

-- 5.- Obtener una estadística de número de pedidos (cuántos) y importe total vendido
por un determinado empleado. Si el empleado no tiene pedidos saldrá una tabla vacía.

-- Si el representante no tiene pedidos saldrá una tabla vacía por lo que podemos
poner un SELECT sin IF y como no nos piden datos del empleado nos sirve Pedidos.

IF object_id('PedidosRep2_5', 'P') IS NOT NULL DROP PROC PedidosRep2_5;
GO
CREATE PROCEDURE PedidosRep2_5 @emple INT
AS
    SELECT rep, COUNT(*) AS [nº de pedidos], SUM(importe) AS [Total vendido]
    FROM pedidos WHERE rep=@emple
    GROUP BY rep;
GO
-- Lo probamos.
EXEC PedidosRep2_5 101 -- Un empleado que tiene varios pedidos
EXEC PedidosRep2_5 111 -- Un empleado que no tiene pedidos
EXEC PedidosRep2_5 200 -- Un empleado que no existe

-- 6.- Añadir a la información obtenida el nombre y nº de oficina del empleado.
-- Necesitamos ahora añadir la tabla de empleados para obtener el nombre del empleado
y su nº de oficina.

-- Si el empleado no tiene pedidos tiene que salir una tabla vacía por lo que los
empleados que no tienen pedidos no tienen que salir → INNER

IF object_id('PedidosRep2_6', 'P') IS NOT NULL DROP PROC PedidosRep2_6;
GO
CREATE PROCEDURE PedidosRep2_6 @emple INT
AS
    SELECT rep, nombre, oficina, COUNT(*) AS [nº de pedidos], SUM(importe) AS
[Total vendido]
    FROM Empleados INNER JOIN pedidos ON numemp=rep
    WHERE rep=@emple

```

```

        GROUP BY rep, nombre, oficina;
GO
-- Lo probamos.
EXEC PedidosRep2_6 101 -- Un empleado que tiene varios pedidos
EXEC PedidosRep2_6 111 -- Un empleado que no tiene pedidos
EXEC PedidosRep2_6 200 -- Un empleado que no existe

-- 7.- Ahora, si el empleado no tiene pedidos a contabilizar que no aparezca la tabla vacía.

-- Hay poner el SELECT dentro de un IF para que salga la tabla únicamente si el empleado tiene pedidos.

IF object_id('PedidosRep2_7', 'P') IS NOT NULL DROP PROC PedidosRep2_7;
GO
CREATE PROCEDURE PedidosRep2_7 @emple INT
AS
    IF EXISTS(SELECT * FROM Pedidos WHERE rep=@emple)
        SELECT rep, nombre, oficina, COUNT(*) AS [nº de pedidos], SUM(importe) AS [Total vendido]
        FROM Empleados INNER JOIN pedidos ON numemp=rep
        WHERE rep=@emple
        GROUP BY rep, nombre, oficina;
GO
-- Lo probamos.
EXEC PedidosRep2_7 101 -- Un empleado que tiene varios pedidos
EXEC PedidosRep2_7 111 -- Un empleado que no tiene pedidos
EXEC PedidosRep2_7 200 -- Un empleado que no existe

-- 8.- Lo mismo pero ahora que aparezca el empleado (su código, nombre y código de oficina) aunque no tenga pedidos.
-- Quitamos pues el IF y como tiene que salir el empleado aunque no tenga pedidos, hay que transformar el INNER en un LEFT y rectificar algunos campos.

IF object_id('PedidosRep2_8', 'P') IS NOT NULL DROP PROC PedidosRep2_8;
GO
CREATE PROCEDURE PedidosRep2_8 @emple INT
AS
    SELECT numemp, nombre, oficina, COUNT(numpedido) AS [nº de pedidos], SUM(importe) AS [Total vendido]
    FROM Empleados LEFT JOIN pedidos ON numemp=rep
    WHERE numemp=@emple
    GROUP BY numemp, nombre, oficina;
GO
-- Lo probamos.
EXEC PedidosRep2_8 101 -- Un empleado que tiene varios pedidos
EXEC PedidosRep2_8 111 -- Un empleado que no tiene pedidos
EXEC PedidosRep2_8 200 -- Un empleado que no existe

-- 9.- Listar los pedidos de un determinado cliente en un determinado mes y determinado año. Sólo queremos el código del cliente y sólo de los clientes que tienen pedidos en el periodo indicado.

IF object_id('PedCliMensual_2_9', 'P') IS NOT NULL DROP PROC PedCliMensual_2_9;
GO
CREATE PROCEDURE PedCliMensual_2_9 @cli INT, @mes INT, @anyo INT
AS
    SELECT * FROM pedidos WHERE clie=@cli AND MONTH(fechapedido) = @mes AND YEAR(fechapedido) = @anyo;
GO
EXEC PedCliMensual_2_9 2103, 5, 2008 -- Un cliente que tiene pedidos en el mes y año indicados
EXEC PedCliMensual_2_9 2103, 5, 2010 -- Un cliente que tiene pedidos en el mes indicado pero no el año indicado
EXEC PedCliMensual_2_9 2103, 4, 2008 -- Un cliente que tiene pedidos en el año indicado pero no del mes indicado

```

```

EXEC PedCliMensual_2_9 2103, 2, 2010 -- Un cliente que tiene pedidos pero ni del mes
ni del año indicados
EXEC PedCliMensual_2_9 2105, 2, 2010 -- Un cliente que no tiene pedidos
EXEC PedCliMensual_2_9 2203, 2, 2010 -- Un cliente que no existe

-- 10.- Ahora el código del cliente deberá aparecer aunque no tenga pedidos en el
período.

-- Tenemos que buscar en todos los clientes aunque los que no tengan pedidos→
necesitamos un JOIN externo.
-- Esta consulta es algo complicada, la he puesto para quien quiera ampliar.

IF object_id('PedCliMensual_2_10', 'P') IS NOT NULL DROP PROC PedCliMensual_2_10;
GO
CREATE PROCEDURE PedCliMensual_2_10 @cli INT, @mes INT, @anyo INT
AS
    SELECT numclie, p.* FROM Clientes LEFT JOIN (SELECT * FROM pedidos WHERE
MONTH(fechapedido) = @mes AND YEAR(fechapedido)= @anyo) AS P ON numclie=clie
    WHERE numclie=@cli
-- ¡Ojo!, tienes que cambiar clie por numclie, recuerda, cuando añades un LEFT o
RIGHT tienes que utilizar los campos de la tabla de la que se sacan todos los
registros.
GO
EXEC PedCliMensual_2_10 2103, 5, 2008 -- Un cliente que tiene pedidos en el mes y
año indicados
EXEC PedCliMensual_2_10 2103, 5, 2010 -- Un cliente que tiene pedidos en el mes
indicado pero no el año indicado
EXEC PedCliMensual_2_10 2103, 4, 2008 -- Un cliente que tiene pedidos en el año
indicado pero no del mes indicado
EXEC PedCliMensual_2_10 2103, 2, 2010 -- Un cliente que tiene pedidos pero ni del
mes ni del año indicados
EXEC PedCliMensual_2_10 2105, 2, 2010 -- Un cliente que no tiene pedidos
EXEC PedCliMensual_2_10 2203, 2, 2010 -- Un cliente que no existe

-- 11.- Rectificar el ejercicio anterior para que salga el texto "el cliente xxx no
existe" en vez de una tabla vacía.
-- Tenemos que añadir un IF.

IF object_id('PedCliMensual_2_11', 'P') IS NOT NULL DROP PROC PedCliMensual_2_11;
GO
CREATE PROCEDURE PedCliMensual_2_11 @cli INT, @mes INT, @anyo INT
AS
    IF NOT EXISTS(SELECT * FROM Clientes WHERE numclie=@cli)
        SELECT 'El cliente ' + STR(@cli,5) + ' no existe'
    ELSE
        SELECT numclie, p.* FROM Clientes LEFT JOIN (SELECT * FROM pedidos WHERE
MONTH(fechapedido) = @mes AND YEAR(fechapedido)= @anyo) AS P ON
numclie=clie
        WHERE numclie=@cli
GO
EXEC PedCliMensual_2_11 2103, 5, 2008 -- Un cliente que tiene pedidos en el mes y
año indicados
EXEC PedCliMensual_2_11 2103, 5, 2010 -- Un cliente que tiene pedidos en el mes
indicado pero no el año indicado
EXEC PedCliMensual_2_11 2103, 4, 2008 -- Un cliente que tiene pedidos en el año
indicado pero no del mes indicado
EXEC PedCliMensual_2_11 2103, 2, 2010 -- Un cliente que tiene pedidos pero ni del
mes ni del año indicados
EXEC PedCliMensual_2_11 2105, 2, 2010 -- Un cliente que no tiene pedidos
EXEC PedCliMensual_2_11 2203, 2, 2010 -- Un cliente que no existe

```

```

-- 12.- Lo mismo que la anterior, pero si no se indica el año se entenderá el actual.

-- Aquí el problema es que el valor por defecto es una expresión (una función que
utiliza el GETDATE())
-- Pero la definición de un valor preterminado en los parámetros no admite
expresiones sólo admite valores fijos.
-- Lo solucionamos asignando al parámetro un valor por defecto concreto que no sea un
valor posible, y luego al empezar el procedimiento cambiamos ese valor por el de la
fecha actual.
IF object_id('PedCliMensual_2_12', 'P') IS NOT NULL DROP PROC PedCliMensual_2_12;
GO
CREATE PROCEDURE PedCliMensual_2_12 @cli INT, @mes INT, @anyo INT = NULL
AS
    IF @anyo IS NULL SET @anyo= YEAR(GETDATE())
    IF NOT EXISTS(SELECT * FROM Clientes WHERE numclie=@cli)
        SELECT 'El cliente ' +STR(@cli,5) + ' no existe' AS [ ]
    ELSE
        SELECT numclie, p.* FROM Clientes LEFT JOIN (SELECT * FROM pedidos WHERE
            MONTH(fechapedido) = @mes AND YEAR(fechapedido)= @anyo) AS P ON
            numclie=clie
        WHERE numclie=@cli
GO
-- Como este procedimiento es una copia del anterior a falta de los parámetros
opcionales, aquí solo bastará probar si los parámetros funcionan
EXEC PedCliMensual_2_12 2103, 5,2008          -- Con todos los parámetros
EXEC PedCliMensual_2_12 2103, 5              -- Con el año por defecto
EXEC PedCliMensual_2_12 2101, DEFAULT, 2008  -- Con el mes por defecto da error el
mes no es opcional
EXEC PedCliMensual_2_12 2103                -- Con el mes y año por defecto da
error falta el mes
-- Para probarlo mejor habría que cambiar el año de un pedido del cliente 2103 para
que tenga un pedido del año actual.

-- Por supuesto podemos refinar estos últimos procedimientos para que escriban un
mensaje si no hay pedidos, etc.

```