INSERT

Podemos añadir con INSERT de dos maneras:

--Indicando los campos y sus valores según el orden

USE compras;

INSERT INTO Articulos (codigart, descrart, preunart, stockart, stockmin, fecbaja) VALUES

('0005', 'Estantería', 300, 5, 1, null);

USE compras;

INSERT INTO Articulos VALUES

('0007', 'MESA LARGA', 800, 7, 1, NULL);

/\* Lo podemos hacer indicándole las columnas que deseemos modificar con sus valores correspondientes \*/

/\* Podemos hacerlo sin indicar los campos, pero tendremos que poner todos los campos, excepto con el campo identity (ver los apuntes) \*/

--Podemos insertar varias filas al mismo tiempo

USE compras;

INSERT INTO Articulos (codigart, descrart, preunart, stockart, stockmin) VALUES

('0008', 'Pelis', 7, 1, 1),

('0009', 'borrador', 4, 1, 1);

/\* Podemos crear una tabla con los campos resultantes de una consulta,

pero antes tenemos que tener creado la nueva tabla. Tiene que coincidir

el tipo de campo con los que vamos a introducir \*/

/\* También lo podemos hacer toda de una sola vez, en la misma consulta

Ejercicio:

En la BD crear una copia de la tabla pedidos \*/

USE jardineria;

SELECT \*

INTO Nueva\_Pedidos

FROM Pedidos;

/\* Crear una nueva tabla ESPAÑA con todas las oficinas que están en España \*/

USE jardineria;

SELECT \*

INTO España

FROM Oficinas

WHERE Pais='España';

/\* Crear una tabla resumen con el código, la descripción de aquellos artículos cuyo código sea menor o igual que 6 \*/

USE compras;

SELECT codigart, descrart

INTO Resumen

FROM Articulos

WHERE codigart <= 6;

/\* Crear una tabla resumen con el código, la descripción de aquellos artículos cuyo código sea menor o igual que 6 \*/

INSERT INTO Resumen

SELECT codigart, descrart

FROM Articulos

WHERE codigart > 6;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Actualiza el equipo de Paul Gasol a los Knicks \*/

UPDATE jugadores SET Nombre\_equipo = 'Knicks'

WHERE Nombre='Pau Gasol';

/\* Actualiza el peso de los jugadores a kilos \*/

UPDATE jugadores SET Peso = Peso\*0.45

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELETE

/\* Crear una nueva tabla EMP\_BCN con los empleados cuya oficina está en la ciudad de BCN \*/

USE jardineria;

SELECT \* INTO EMP\_BCN FROM Oficinas

WHERE Ciudad='Barcelona';

USE jardineria;

SELECT \* INTO Empleados\_Barcelona FROM Empleados

WHERE CodigoOficina IN (SELECT CodigoOficina FROM Oficinas

WHERE Ciudad='Barcelona');

/\* Insertar un nuevo cliente con el límite de crédito = 0.

Luego borrarlo a través de otra consulta con la condición de límite de crédito = 0 \*/

USE jardineria;

INSERT INTO Clientes VALUES (40, 'Roberto', 'Lopez', 'Goldfish', '123456789', '123456789',

'False Street 52 2 A', 'False Street 52 2 A', 'Fuenlabrada', 'Madrid', 'España', '26005', 8, 0);

/\* Borrar el Cliente anterior a través de otra consulta con la condición de límite de crédito = 0 \*/

DELETE FROM Clientes

WHERE CodigoCliente IN

(SELECT CodigoCliente FROM Clientes

WHERE LimiteCredito = 0);

/\* Borrar aquellos Empleados de la tabla Empleados cuya oficina este en Barcelona \*/

DELETE FROM Empleados WHERE CodigoOficina IN (SELECT CodigoOficina FROM Oficinas

WHERE Ciudad='Barcelona');

Mens. 547, Nivel 16, Estado 0, Línea 2

Instrucción DELETE en conflicto con la restricción REFERENCE "Clientes\_EmpleadosFK". El conflicto ha aparecido en la base de datos "jardineria", tabla "dbo.Clientes", column 'CodigoEmpleadoRepVentas'.

Se terminó la instrucción.

/\* Borrar aquellos Empleados de la tabla Empleados cuya oficina este en Barcelona y además el código del empleado no este\*/

DELETE FROM Empleados WHERE CodigoOficina IN (SELECT CodigoOficina FROM Oficinas

WHERE Ciudad='Barcelona') AND CodigoEmpleado NOT IN

(SELECT CodigoEmpleadoRepVentas FROM Clientes);

---------------------------------------------------------------------------

08/01/2015

--crear una nueva tabla de los empleados que sean representante de ventas que no tengan clientes

--borrar de la tabla empleados los empleados que no tengan clientes y que sean representantes de ventas  
--insertar todos los empleados de la tabla RepreSinClientes de vuelta a la tabla empleados  
--insertar un registro nuevo en la tabla equipos  
--eliminar todos los equipos que no hayan jugado partidos como locales, resolviendo con join  
usenba

09/01/2015

Use nba;

SELECT Nombre, equipo\_local FROM equipos

LEFT JOIN partidos ON

Nombre=equipo\_local WHERE Nombre='CAI';

DELETE FROM equipos

WHERE Nombre IN (SELECT Nombre FROM equipos

LEFT JOIN

--Borrar los equipos de la tabla equipos que nunca hayan jugado un partido (insertar un registro)

CASE

--Listar los artículos mostrando su precio categorizado por alto, bajo y no establecido.

SELECT descrart, preunart, 'precio' =

CASE

WHEN preunart IS NULL THEN 'No establecido'

WHEN preunart < 200 THEN 'Bajo'

ELSE 'Alto'

END

FROM articulos

ORDER BY precio

-- Actualizar los artículos con precio < 10 con 5% y los > =10 con un 7%

UPDATE Articulos SET preunart =

CASE

WHEN preunart < 10 THEN preunart \* 1.05

ELSE preunart \* 1.07

END

-- Una consulta de seleccion. En una columna ganador el ganador de todos los partidos (equipo) de la temporada 98-99

-- si hay empate que salga 'empate'

USE nba;

SELECT 'EQUIPO\_GANADOR' =

CASE

WHEN puntos\_local > puntos\_visitante THEN equipo\_local

WHEN puntos\_local < puntos\_visitante THEN equipo\_visitante

ELSE 'EMPATE'

END

FROM partidos WHERE temporada = '98/99';

-- En la tabla jardineria modificar el precio de los productos tal que se obtenga como minimo un 20% de

-- beneficio (por precio unitario) si el producto ya esta a un 20% de beneficio o mas no tenemos que modificar nada.

USE jardineria;

SELECT PrecioVenta =

CASE

WHEN PrecioVenta-PrecioProveedor < PrecioVenta\*0.20

THEN PrecioVenta\*1.2

ELSE PrecioVenta

END

FROM Productos

/\*

Valorar los clientes según los pedidos realizados

< 3 = regular

between 3 and 4 = normal

> 4 = bueno

\*/

Use jardineria;

SELECT CodigoCliente, COUNT (CodigoCliente) Cantidad\_pedido, 'Valor\_Cliente' =

CASE

WHEN COUNT (CodigoCliente) < 3 THEN 'Regular'

WHEN COUNT (CodigoCliente) BETWEEN 3 AND 4 THEN 'Normal'

WHEN COUNT (CodigoCliente) > 4 THEN 'Bueno'

-- ELSE 'Bueno' (OPCIONAL)

END

FROM Pedidos

GROUP BY CodigoCliente;

-- Ahora incluyendo el nombre del cliente

Use jardineria;

SELECT C.NombreCliente, P.CodigoCliente, COUNT (P.CodigoCliente) Cantidad\_pedido, 'Valor\_Cliente' =

CASE

WHEN COUNT (P.CodigoCliente) < 3 THEN 'Regular'

WHEN COUNT (P.CodigoCliente) BETWEEN 3 AND 4 THEN 'Normal'

WHEN COUNT (P.CodigoCliente) > 4 THEN 'Bueno'

-- ELSE 'Bueno' (OPCIONAL)

END

FROM Pedidos P, Clientes C

WHERE P.CodigoCliente=C.CodigoCliente

GROUP BY P.CodigoCliente, C.NombreCliente

ORDER BY Cantidad\_pedido;

Tarea hacer el ejercicio del IVA con un CASE

Modifica la tabla DetallePedido para insertar un campo numérico llamado IVA (hacerlo gráficamente).

Establece el valor de ese campo a 18 para aquellos registros cuyo pedido tenga fecha a partir de Julio 2010 y el resto de pedidos estableciendo el IVA al 21.

UPDATE DetallePedidos SET IVA=18

WHERE CodigoPedido IN (SELECT CodigoPedido FROM Pedidos

WHERE FechaPedido >='2010-07-01');

UPDATE DetallePedidos SET IVA=16

WHERE CodigoPedido IN (SELECT CodigoPedido FROM Pedidos

WHERE FechaPedido <'2010-07-01');

Con un CASE

USE jardineria;

UPDATE DetallePedidos SET IVA =

CASE

WHEN CodigoPedido IN

(SELECT CodigoPedido FROM Pedidos

WHERE FechaPedido >= '2010-07-01') THEN 18

ELSE 21

END;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Tema nuevo: Creación de datos