Consultas SQL

Creacion de una tabla temporal

CREATE TEMPORARY TABLE temporal (id int(11), name varchar(40));

CREATE TEMPORARY TABLE temporal (id int(11), name varchar(40));

INSERT INTO temporal (id, name) SELECT id\_USer, concat(first\_name," ",last\_name) FROM usuarios WHERE usuarios.id\_User=1;

select \* FROM temporal;

Insertar datos desde usuarios hasta la cabecera de albaranes. directamente

INSERT INTO head\_alb (id\_USer, nom\_User, email\_user) SELECT id\_USer, concat(usuarios.first\_name," ",usuarios.last\_name), usuarios.email\_Adrees\_User FROM usuarios WHERE usuarios.id\_User=1;

Añadiendo datos a unas variables

SELECT @id:= usuarios.id\_User, @nombre:= concat(usuarios.first\_name," ",usuarios.last\_name), @mail:= usuarios.email\_Adrees\_User from usuarios WHERE usuarios.id\_User=1;

Insertando datos desde variables

INSERT INTO head\_alb (id\_USer, nom\_User, email\_user) SELECT @id, @nombre, @mail;

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

CREATE TEMPORARY TABLE temporal\_articul (idArt int(11), conceptoArt varchar(50), precio\_BaseArt decimal(5,2), ivaArt decimal(3,2));

insert into temporal\_articul (idArt, conceptoArt, precio\_BaseArt, ivaArt ) SELECT id\_Art, concepto, precio\_Base, tipo\_Iva from articul where articul.id\_Art=1;

SELECT @cantTemp := 15; /\*Asignación de un valor fijo a una variable\*/

SELECT @temAlb := MAX(id\_Alb) from head\_alb; /\*Asignación del Valor Maximo del campo id\_Alb de la tabla head\_alb\*/

SELECT @temAlb := @temAlb + 1; /\*Sumar una posición más al valor de la variable que se obtine de la tabla head\_alb\*/

CREATE TEMPORARY TABLE temporal\_user (id int(11), name varchar(40), mail varchar(50)); /\*Creación de una tabla temporal para los usuarios\*/

INSERT INTO temporal\_user (id, name, mail) SELECT id\_USer, concat(first\_name," ",last\_name), email\_Adrees\_User FROM usuarios WHERE usuarios.id\_User=1;

/\*Injección de datos de la tabla usuarios a la tabla temporal de usuarios \*/

CREATE TEMPORARY TABLE temporal\_articul (temAlb int(11), idArt int(11), conceptoArt varchar(50), precio\_BaseArt decimal(5,2), ivaArt decimal(3,2));

/\*Creación de una tabla temporal para articulos\*/

INSERT INTO temporal\_articul (temAlb, idArt, conceptoArt, precio\_BaseArt, ivaArt) SELECT @temAlb, id\_Art, concepto, precio\_Base, tipo\_Iva from articul where articul.id\_Art = 5;

/\*Injección de los datos basico de la tabla articulos a la tabla temporal de articulos\*/

set @idArtemp =(SELECT temporal\_articul.IdArt from temporal\_articul);

/\*Asignacion de Id del articulo a la variable @idArtemp\*/

select \* from temporal\_user;

/\*Muestra los datos que hay en la tabla temporal\_user\*/

select idArt, conceptoArt, precio\_BaseArt, ivaArt from temporal\_articul;

/\*Muestra los datos que hay en la tabla temporal\_articul\*/

set @temprecioBase =(SELECT temporal\_articul.precio\_BaseArt from temporal\_articul where temporal\_articul.idArt=@idArtemp);

/\*Declaración de la variable temprecioBase y asignación del valor precio base de la tabla temporal de articulo, usando como referencia el Idarticulo \*/

set @tempValorIva =(SELECT temporal\_articul.ivaArt from temporal\_articul where temporal\_articul.idArt=@idArtemp);

/\*Declaracion de la variable tempValorIva y asignacion de la cuota de iva para el calculo de la linea\*/

SELECT @temprecioBase, @tempValorIva;

/\*Muestra los valores que se ha asignado a las variables @temprecioBase y @tempValorIva\*/

SELECT @temLinea := @temprecioBase \* @cantTemp;

/\*Calculo con la variable cantidad multiplicada por el valor base del articulo\*/

SELECT @temIva := (@tempValorIva \* @temLinea) - @temLinea;

/\*Calculo de la cantidad de iva partiendo de los valores obtenidos de las tablas\*/

SELECT @temTotalLinea := @temLinea + @temIva;

/\*Declaracion de la variable para el total de la linea y asignacion de calculo apartir de los valores de la tabla temporal de articulos y la variable del iva\*/

INSERT INTO head\_alb (id\_USer, nom\_User, email\_user) SELECT id, name, mail from temporal\_user where temporal\_user.id = 1;

/\*Inyección de los datos a la tabla head\_alb desde la tabla temporal de usuarios\*/

INSERT INTO body\_alb (id\_Alb, id\_Art, concepto, cantidad, precio\_Base, tipo\_Iva, cuota\_iva, precio\_Total\_Linea) SELECT @temAlb, idArt, conceptoArt, @cantTemp, precio\_BaseArt, ivaArt, @temIva, @temTotalLinea from temporal\_articul WHERE temporal\_articul.temAlb = @temAlb or temporal\_articul.temAlb = 0;

/\*Inyección de los datos de la linea en la tabla body\_alb a partir de la tabla temporal de articulos y los calculos de iva y total linea\*/

drop TABLE temporal\_user;

drop TABLE temporal\_articul;

/\*eliminación de las tablas temporales para usuarios y articulos\*/

UPDATE desde BBDD Distinas

Estaba en la BBDD de Poblaciones

update zcStocks.zcAlmacenes set idPoblacion = 1 where zcStocks.zcAlmacenes.idAlmacen = 1 ;  
select \* from zcStocks.zcAlmacenes;

Select entre BBDD

select po.idPoblacion as IdPoblacion, po.nombrePoblacion as Poblacion, po.cpPoblacion as CP, al.codAlmacen as CodigoAlmacen, al.descripcion as Descripcion, al.direccion as Direccion from zcPoblaciones.zcPoblacion po join zcStocks.zcAlmacenes al on po.idPoblacion = al.idPoblacion

Aqui se ve que estoy uniendo la tabla Poblacion de la BBDD zcPoblaciones con la Tabla zcAlmacen de la BBDD zcStocks y recupero de zcPoblacion la id, nombre, CP, y de zcAlmacen el codigo del Alamcen, Descripcion, direccion, (OJO este consulta es inter BBDD zcStock y zcPoblaciones).

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

Inner Join entre la tabla de Clientes y la de delegaciones

select Clientes.idCli as CodigoCli, nombreCli as Nombre, Clientes.direccion as Direccion, Clientes.telefono as Telefono, Delegacion.direccion as DirecDelegacion from Clientes inner join Delegacion on Clientes.idCli = Delegacion.idCli order by Clientes.idCli;



update OfertaCabecera o left join Delegacion d on o.idCli = d.idCli set o.idDelega = d.idDelega, o.direcDelega=d.direccionDelega;

CREATE TEMPORARY TABLE temporal\_user (id int(11), name varchar(40), mail varchar(50));

insert into temporal\_user (id, name, direccion, telefono) select idCli, nombreCli, direccion, telefono from Clientes where Clientes.idCli = 3;

Inyeccion de datos a la tabla OfertaCabecera desde las tablas Clientes y Delegaciones.

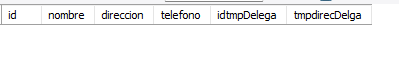
Por que lo hago de esta manera, pq me ha dado cuenta que puedo crear una tabla temporal con tantos campos como necesite y aquí ir inyectando datos de distintas tabla y despues pasarlo todo a la tabla que me interese.

Parto de este suceso y no realizo join on anidados pq inserto todo un nuevo registro en la tabla destino.

creo la variable SELECT @tmpcli := xx (‘XX se ha de sustituir por un numero’) para almacenar el numero de cliente que quiero inyectar

Creo una tabla temporal con todos los campos que necesito tanto de la tabla clientes como de la tabla delegación.

create temporary table tmpfision (id int(11), nombre varchar(30), direccion varchar(30), telefono varchar(12), idtmpDelega int(11), tmpdirecDelga varchar (50));



inserto desde la tabla clientes datos usando como referencia la variable temporal

insert into tmpfision (id, nombre, direccion, telefono) select idCli, nombreCli, direccion, telefono from Clientes where Clientes.idCli = @tempcli;



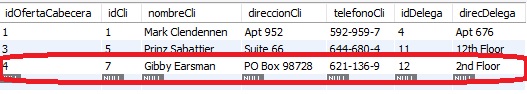
actualizo los campos de la tabla temporal que tengo en null con los datos de la tabla delegacion.

update tmpfision o left join Delegacion d on o.id = d.idCli set o.idtmpDelega = d.idDelega, o.tmpdirecDelga=d.direccionDelega;



inyecto todos los datos en la para la cabecera de la oferta usan como condicion la variable temporal

insert into OfertaCabecera (idCli, nombreCli, direccionCli, telefonoCli, idDelega, direcDelega) select id, nombre, direccion, telefono, idtmpDelega, tmpdirecDelga from tmpfision where tmpfision.id = @tempcli;



Elimino la tabla temporal

drop table tmpfision

@tmpOferta :=3;

@tmpArt :=179;

insert into tmpArt (idArt, desArt, precio, idIva) select idArt, desArticulo, pvp, idIva from Articulos where idart = @tmpArt;

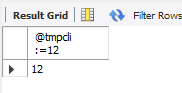
create temporary table tmplineas (idOferta int(11), idArt int(11), desArt varchar (30), pvp double (6,2), idIva int(11), tipoIva double (6,2), totalLinea double (8,2));

insert into tmplineas (idOferta) select idOfertaCabecera from OfertaCabecera where idOfertaCabecera = @tmpOferta;

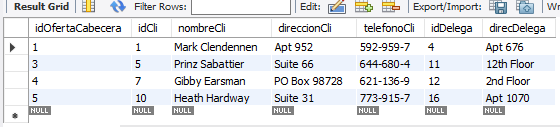
Modelo Consulta SQL Directo.

Primero crear una variable en la que asignare el valor del código de cliente.

select @tmpcli :=12

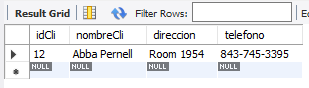


El las cabeceras de Ofertas no tenemos ninguna con Cliente 12 voy a añadirlo.



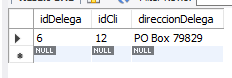
Veo con el siguiente select cuales son los datos del clientes con código 12

Select \* from Clientes where idCli = @tmpcli



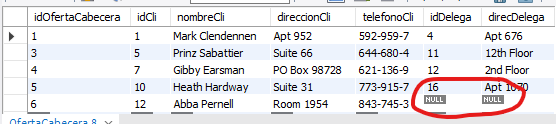
Ahora voy a ver que delegaciones tiene el cliente 12

Select \* from Delegacion where idCli =@tmpcli



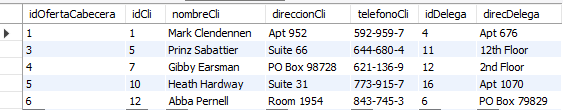
Ahora voy añadir los datos de Cliente y Delegación a la tabla OfertaCabecera.

insert into OfertaCabecera (idCli, nombreCli, direccionCli, telefonoCli ) select idCli, nombreCli, direccion, telefono from Clientes where Clientes.idCli = @tmpcli;



Ahora los datos de la delegación.

update OfertaCabecera o left join Delegacion d on o.idCli = d.idCli set o.idDelega = d.idDelega, o.direcDelega = d.direccionDelega;



Para rellenar las líneas de Ofertas

Declaración de variables temporables que serán datos obtenidos a traves de las pantallas y opciones del erp.

select @tmpoferta :=6; La Oferta

select @tmpart :=29; El Articulo

select @tmpcode :=0;

select @tmpiva :=0;

select @tmpvp :=0;

select @tmprow :=0;

select @tmpcuota :=0;

select @tmptotalinea :=0;

A ver que campos tengo

Select \* from OfertasLineas



Necesito idOferta para ver que cabecera el corresponde.

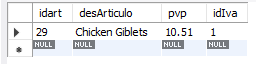
Select @tmpoferta = 6;

Declaro la variable que obtendría de como código de articulo

Select @tmpart :=29;

Que articulo es el 29

select \* from Articulos where idArt = @tmpart;

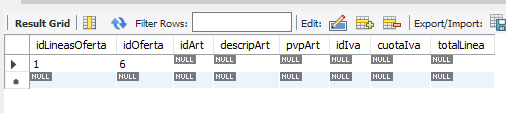


Añado la referencia de la cabecera a la línea

insert into OfertasLineas (idOferta) select idOfertaCabecera from OfertaCabecera where OfertaCabecera.idOfertaCabecera = @tmpoferta;

El registro 1 tiene la cabecera 6 de Oferta.

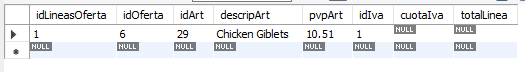
Select \* from OfertasLineas



Ahora voy a añadir a la línea el articulo.

Detacar que el left join on lo comparo con la variable temporal del articulo.

update OfertasLineas o left join Articulos a on a.idart = @tmpart set o.idArt = a.idart, o.descripArt = a.desArticulo, o.pvpArt = a.pvp, o.idIva = a.idIva;



set @tmpiva = (select tipo from Iva where idIva = 1);

El id del iva lo obtendre de la tabla del articulo.

Bien para calcular los valores de cuotaIva y totalLinea voy a usar el idLineasOferta ya que este no se repite.

Set @tmprow = (select idLineasOferta from OfertasLineas where idOferta);

ahora se que con @tmprow = 3 y me servira para obtener el code del iva del articulo que estoy trabajando.

Set @tmpcode = (select idIva from OfertasLineas where idLineasOfertas = @tmprow and idOferta = @tmpoferta and idArt = @tmpart);

Con esta consulta obtengo el codigo del tipo de Iva. Se plantea de esta forma para que si en el futuro el tipo de iva cambiase, este pueda ser actualizado en la tabla de iva sin que esto perjudique las facturas ya generadas.

Obtención del precio de venta de un producto.

Set @tmpvp = (select pvpArt from OfertasLineas where idLineasOferta = @tmprow and idOferta = @tmpoferta and idArt = @tmpart);

@tmpvp = 10.51

obtener el tipo de iva.

Set @tmpiva =(select tipo from iva where idIva = @tmpcode);

@tmpiva = 1.21

ya tengo todos los elementos necesarios para poder complementar los campos de la cuota de iva de la linea y el total del articulo en la lineas (cuotaIva and totalLinea).

El total del producto.

Set @tmptotalinea = (@tmpvp \* @tmpiva)

La cuota de IVA

set @tmpcuota = (@tmptotalinea - tmpvp)

actualización de la linea del articulo con la cuota de iva y el total del articulo en la lineas

update OfertasLineas set cuotaIva = @tmpcuota, totalLinea = @tmptotalinea where idLineasOferta = @tmprow and idOferta = @tmpoferta and idArt = @tmpart;