



# Trabajo Practico Integrador De Base de datos 1

**Tema :** Compras de clientes en MY SQL S.A

**Integrantes :** Alejandro Quevedo,  
Gonzalo Gomez,  
Florencia Alvarez,  
Bernardette Aufdermaur,  
Damian Rosa

**Fecha de Entrega :** 06/12/2014

**Profesores :** Ruben Dario Moreira,  
Alejandro Goitea,  
Jose Luis Cabrera.

**Achivos :** CompraTp.sql, Der.Jpeg



**1- Realizar el DER del siguiente enunciado.**

La empresa "My SQL SA" necesita registrar las compras que realizan los diversos clientes en la empresa.

La compra de productos, resulta la operación más importante de la empresa.

Cuando el cliente efectúa la compra no necesariamente se abonará en el momento, sino que podría retrasarse. Solo en el momento de abonar las facturas

se dará por finalizada la compra.

Cada factura siempre hará referencia a una compra realizada.

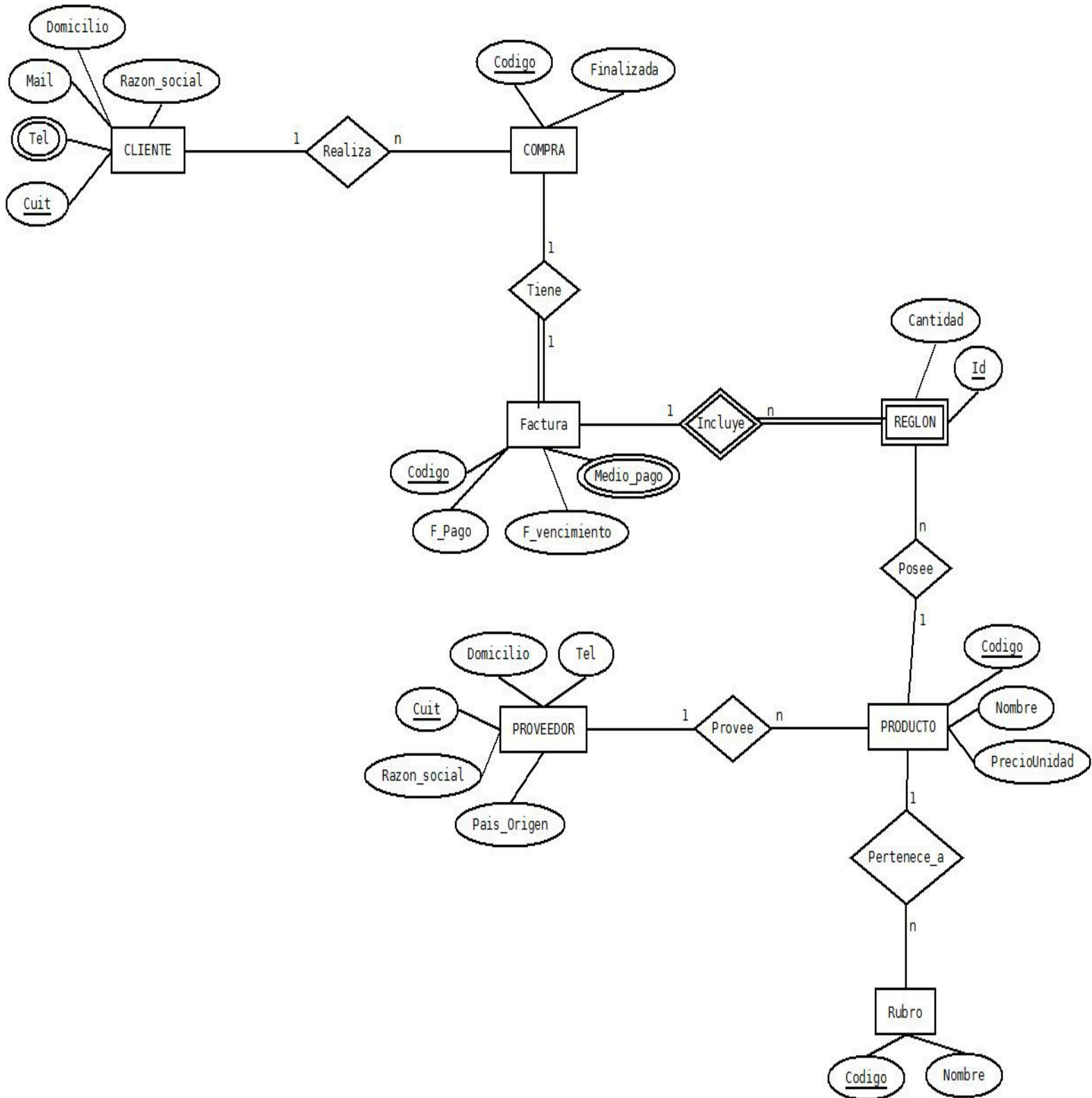
Toda factura número de factura, una fecha de vencimiento, fecha de pago y el medio de pago (Efectivo, Tarjeta, Cheque).

Los cliente se identificará unívocamente. Por cada uno de estos se desea almacenar la razón social, el CUIT, los datos domiciliarios, puede tener varios números de teléfono y una dirección de mail. Una compra es efectuada por un único cliente.

Los productos que comercializa la empresa pueden ser de diferentes rubros. De cada uno de los productos se almacenará el código unívoco, nombre, precio actual, y proveedor.

Por cada compra se necesitan conocer los productos y, por cada uno de ellos, la cantidad de unidades.

De los proveedores, también se almacenará la Razón social, CUIT y datos Domiciliarios, teléfono, país de origen.





## 2 – Pasar el DER a MR.

CLIENTE(Cuit, Razon\_Socail,Mail,Domiciolio)  
PK Cuit

PROVEEDOR(Cuit,Razon\_Social,Domicilio,pais\_origen,tel)  
PK Cuit

Rubro(Codigo,,nombre)  
PK codigo

COMPRA(Codigo, Finalizada, Codigo\_cliente)  
PK Codigo  
FK Codigo\_cliente

FACTURA(Codigo, F\_pago,F\_vencimiento,Codigo\_compra)  
PK Codigo  
FK Codigo\_compra

PRODUCTO(Codigo,Nombre,PrecioUnidad,Stock,Codigo\_proveedor)  
PK Codigo  
FK Codigo\_Proveedor

REGLON(Id,Codigo\_factura, Codigo\_producto,cantidad)  
PK Id, Codigo\_factura, Codigo\_producto  
FK Codigo\_factura, Codigo\_producto

Tel\_Cliente(Codigo\_cliente, tel)  
PK Codigo\_cliente, tel  
FK Codigo\_cliente

Medio\_Pago\_Factura(Codigo\_factura, medio\_pago)  
PK Codigo\_factura, medio\_pago  
FK Codigo\_factura

## 3- Verificar si esta normalizado

Se verifica que esta en 1<sup>er</sup> forma normal porque cada atributo es atómico , indivisible, que no pueden tener mas de un único valor, no son multivaluados, tampoco se verifican tuplas idénticas.

Se verifica que esta en 2da forma normal porque esta en 1<sup>er</sup> forma normal y aparte tiene dependencia funcional total osea que cualquier atributo depende completamente de la clave primaria.

Se verifica que esta en 3<sup>er</sup> forma normal porque esta en 2da forma normal, y no tiene dependencia funcional transitiva, osea todos los atributos que no sean claves dependen de la clave completa de forma no transitiva.



#### 4- Crear la base de datos CompraTp, crear cada tabla correspondiente ingresando datos en cada tabla y hacer las siguientes consultas.

Se verifica creación de tablas e inserción de datos en las mismas en el archivo CompraTp.sql

Las consultas son las siguientes:

1- Obtener el total gastado por cada cliente en cada factura

```
select c.razon_social,sum(r.cantidad * p.preciounidad)
from (((cliente c left join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on cm.codigo = f.codigo_compra)
inner join renglon r on f.codigo = r.codigo_factura)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura;
```

2- Obtener el cliente con mayor monto gastado, debe figurar razon social, numero de compra y codigo

```
select c.razon_social as Cliente, cm.codigo as Numerodecompra,sum(r.cantidad *
p.preciounidad) as Monto
from (((cliente c inner join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on f.codigo_compra = cm.codigo)
inner join renglon r on r.codigo_factura = f.codigo)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura
having Monto = ( select max(totales.total) from
( select sum(r.cantidad * p.preciounidad) as total
from (((cliente c inner join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on f.codigo_compra = cm.codigo)
inner join renglon r on r.codigo_factura = f.codigo)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura) as totales );
```

3- Obtener el cliente con menor monto gastado, debe figurar razon social, numero de compra y codigo

```
select c.razon_social as Cliente, cm.codigo as Numerodecompra,sum(r.cantidad *
p.preciounidad) as Monto
from (((cliente c inner join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on f.codigo_compra = cm.codigo)
inner join renglon r on r.codigo_factura = f.codigo)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura
having Monto = ( select min(totales.total) from
( select sum(r.cantidad * p.preciounidad) as total
from (((cliente c inner join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on f.codigo_compra = cm.codigo)
inner join renglon r on r.codigo_factura = f.codigo)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura) as totales );
```



4- Listar codigo, nombre y cantidad del producto mas vendido

```
select p.codigo as CodigoProducto, p.Nombre as Nombre, sum(r.cantidad) as Cantidad
from producto p left join renglon r on r.codigo_producto = p.codigo
group by r.codigo_producto
having sum(r.cantidad) = ( select max(totales.total) from
( select sum(r.cantidad) as total
from producto p left join renglon r on r.codigo_producto = p.codigo
group by r.codigo_producto ) as totales );
```

5 -Listar nombre de cliente, Domicilio, Numero de compra, numero de factura, de cliente con compras mayor a \$3000

```
select c.razon_social as Cliente ,c.domicilio as Domicilio, cm.codigo as Compra,
f.codigo as NumeroFactura,sum(r.cantidad * p.preciounidad) as Monto
from (((cliente c inner join compra cm on c.cuit = cm.cuit_cliente)
inner join factura f on f.codigo_compra = cm.codigo)
inner join renglon r on r.codigo_factura = f.codigo)
inner join producto p on p.codigo = r.codigo_producto
group by r.codigo_factura
having sum(r.cantidad * p.preciounidad) > 3000;
```

6- Listar cantidad de prodcto que vendio cada proveedor

```
select pr.razon_social as RazonSocial, sum(cantidad) as Cantidad
from (proveedor pr left join producto p on p.cuit_proveedor = pr.cuit)
inner join renglon r on r.codigo_producto = p.codigo
group by pr.razon_social;
```

7- Listar proveedor que vendio menos productos, indicando razon social y cantidad de productos vendidos

```
select pr.razon_social as Proveedor, sum(cantidad) as Cantidad
from (proveedor pr left join producto p on p.cuit_proveedor = pr.cuit)
inner join renglon r on r.codigo_producto = p.codigo
group by pr.razon_social
having sum(cantidad) = ( select min(totales.total)
from
(select sum(cantidad) as total
from (proveedor pr left join producto p on p.cuit_proveedor = pr.cuit)
inner join renglon r on r.codigo_producto = p.codigo
group by pr.razon_social) as Totales );
```

8- Listar las compras no finalizadas, indicando a que cliente pertenece, su domicilio y el numero de compra

```
select c.razon_social as Cliente, c.domicilio as Domicilio, cm.codigo as NumeroDeCompra
from cliente c inner join compra cm on c.cuit = cm.cuit_cliente and cm.finalizada = 'no';
```

9- El cliente 30-12347654-2 finalizo su compra

```
Update compra
set finalizada = 'si'
where cuit_cliente = '30-12347654-2';
```



10- Se verifica que la compra 1 ah sido saldada

```
Update compra  
set finalizada = 'si'  
where codigo = 1;
```

11- Se verifica que la empresa ya no seguira vendiendo producto con rubro Bebidas.

```
delete from producto  
where codigo_rubro = ( select codigo from rubro  
                      where nombre = 'Bebidas'  
                      );
```