

# UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMATICA Y CIENCIAS DE  
LA COMPUTACIÓN



Base de datos I

Mini Proyecto

Alumno:  
Diego Varas Moya.  
Profesor:  
Andrea Rodríguez.  
Ayudante:  
Matías Medina

24 de oct. De 2016

# Introducción

Del enunciado dado se construirá una base de datos, para lo cual se deberá realizar un modelo entidad relación para luego obtener un modelo relacional, de donde este ultimo será posible formular respuestas a preguntas dadas en el enunciado usando algebra relacional, para luego pasar nuestro modelo al lenguaje SQL y de ahí poder contestar las preguntas asociadas ya en este modelo final.

# Enunciado

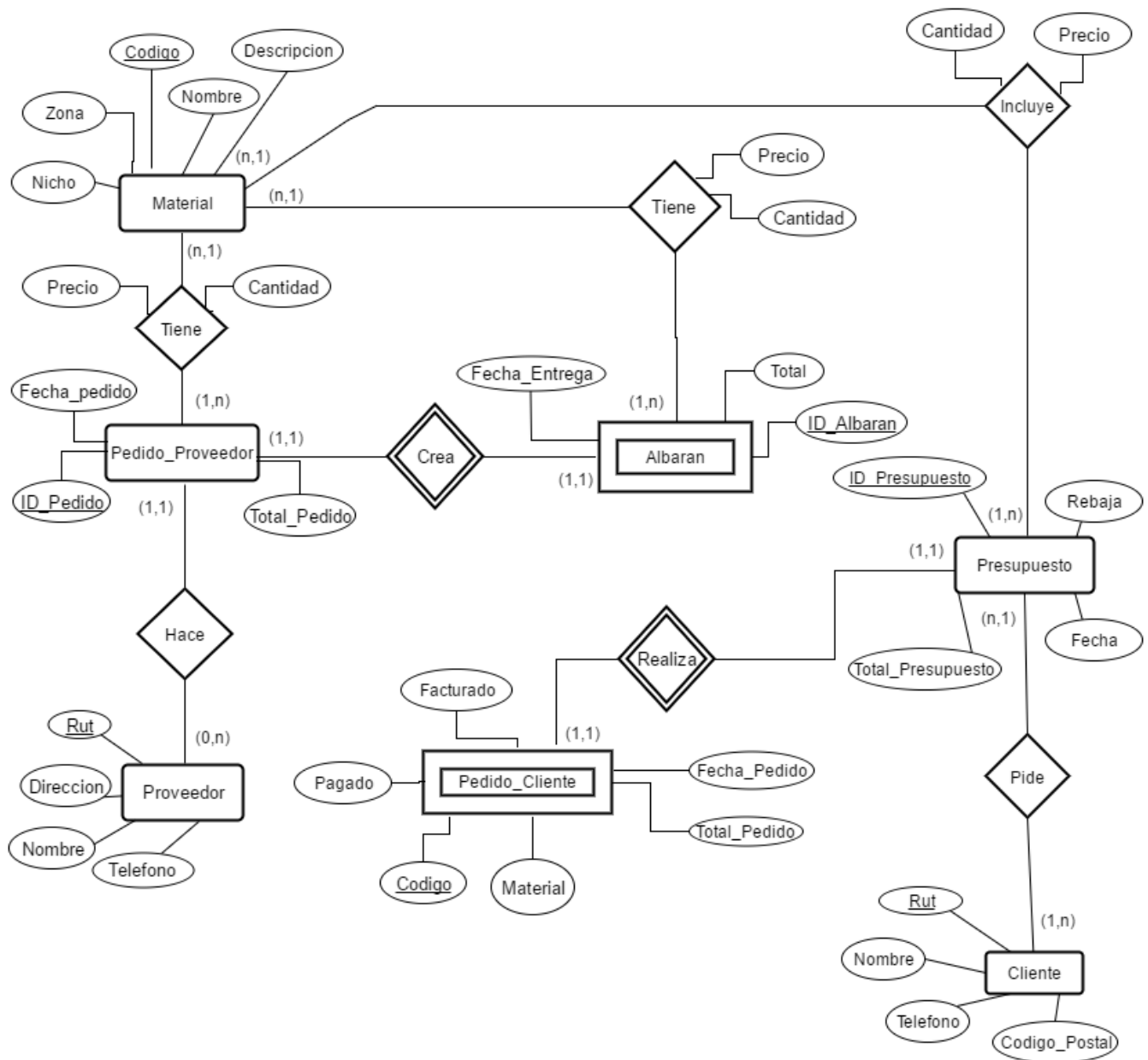
Una compañía de materias de construcción quiere hacer que su sistema de ventas será autónomo. Para aquello, deciden contratar sus servicios y, tras las reuniones iniciales, se consiguen los primeros requisitos del sistema.

Primero, los materiales manejados pueden ser muy diversos, no existiendo una clasificación muy clara entre ellos. Sin embargo, la compañía le pide mantener información de todos ellos, aunque todavía no se haya realizado ninguna compra y no hayan existencias en el almacén. La información de partida a considerar para el material es su código, nombre y descripción. Cuando se recibe el material se sitúa en la zona y nicho que le corresponde, pudiendo haber en el mismo nicho varios materiales de características similares.

En relación con los proveedores se dispone de muchos de ellos, a los cuales se debe incluir en el sistema ya que de ellos se comenzará a recibir con frecuencia la información sobre los productos que ofrecen y los precios actuales de esos productos, precios que se considerarán correctos hasta que llegue nueva información con nuevos precios. Es importante a la hora de hacer un pedido guardar el precio actual del material solicitado para contrastarlo cuando llegue el albarán. La información relativa a los proveedores es su rut, Nombre, Dirección y teléfono. En cuanto al pedido interesa conocer el código de identificación del mismo, los materiales solicitados y la cantidad de cada uno. Una vez realizado el pedido o pedidos al proveedor o proveedores, el material llegará acompañado de un albarán que indica los materiales servidos. Por problemas de distribución es posible que un pedido llegue en varias entregas distintas e incluso que una entrega agrupe materiales solicitados en varios pedidos distintos. En el albarán figura para cada artículo a qué pedido corresponde y el precio, lo que permite determinar si respetaron el precio que figuraba en el pedido.

El sistema debe poder tratar con los clientes, atender sus peticiones, facturarlos y hacer presupuestos. La información de interés del cliente es su Rut, nombre, dirección, código postal y posibles teléfonos de contacto. Un presupuesto refleja siempre los materiales que ha solicitado el cliente con un precio. El pedido del cliente, que siempre se corresponde con un presupuesto previo, se paga con posterioridad a su realización, siendo condición indispensable para la posterior entrega de los materiales. Del pedido del cliente se necesita conocer además, su número de pedido.

# Modelo Entidad Relación



## Restricciones

- Fechas posteriores a la fecha actual
- Precios Mayores que 0
- Cantidades mayores o iguales que 1
- Fecha de entrega posterior a fecha de pedido
- Fecha de pedido posterior a fecha de presupuesto

# Modelo Relacional

Modelo relación a base del Modelo Entidad Relación anterior.

material (código\_material:string, nombre:string, descripción:string, zona:string, nicho:string)

proveedor (rut\_p:string, nombre:string, dirección:string, teléfono:string)

pedido\_proveedor (id\_pedido:string, rut\_p:string, fecha\_pedido:string, total\_pedido:integer)  
FK rut\_p FROM PROVEEDOR (rut\_p)

albarán (id\_albaran:string, id\_pedido:string, fecha\_entrega:string, total:integer)  
FK id\_pedido FROM PEDIDO\_PROVEEDOR (id\_pedido)

pedido\_prov\_material (id\_pedido:string, código\_material:string, cantidad:integer, precio:integer)  
FK id\_pedido FROM PEDIDO\_PROVEEDOR (id\_pedido)  
FK código\_material FROM MATERIAL (código\_material)

albarán\_material (id\_albaran:string, código\_material:string, cantidad:integer, precio:integer)  
FK id\_albaran FROM ALBARAN (id\_albaran)  
FK código\_material FROM MATERIAL (código\_material)

cliente (rut\_c:string, nombre:string, dirección:string, teléfono:string, código\_postal:string)

presupuesto (id\_presupuesto:string, rut\_c:string, total\_presupuesto:integer, rebaja:string, fecha\_presupuesto:string)  
FK rut\_c FROM CLENTE (rut\_c)

pedido\_cliente (código:string, id\_presupuesto:string, total\_pedido\_c:integer, fecha\_pedido\_c:string, pagado:bit, facturado:bit)  
FK id\_presupuesto FROM PRESUPUESTO (id\_presupuesto)

pres\_incluye\_mat (código\_material:string, id\_presupuesto:string, cantidad:integer, precio:integer)  
FK código\_material FROM MATERIAL (código\_material)  
FK id\_presupuesto FROM PRESUPUESTO (id\_presupuesto)

# Consultas en Algebra Relacional

- Nombre de los proveedores que han suministrado hormigón armado a la Compañía desde el 1 de enero de 2000.

R1  $\leftarrow$  pedido\_proveedor **JOINnat** pedido\_prov\_material

R2  $\leftarrow \sigma$  Nombre = 'hormigón armado' (material)

R3  $\leftarrow$  R1 **JOINnat** R2

R4  $\leftarrow \sigma$  Fecha  $\geq$  01/01/2000 (R3)

R5  $\leftarrow \Pi$  rut\_p (R4)

R6  $\leftarrow$  R6 **JOINnat** Proveedor

$\leftarrow \Pi$  Nombre (R6)

- Total de unidades (metros cuadrados) de pavimento de gres vendidas en el último mes.

R1  $\leftarrow$  pres\_incluye\_mat **JOINnat** pedido\_cliente

R2  $\leftarrow$  R1 **JOINnat** material

R3  $\leftarrow \sigma$  Nombre = 'pavimento de gres' (R2)

R4  $\leftarrow \sigma$  fecha\_pedido\_c  $>$  F Actual - 1 mes (R3)

R5  $\leftarrow \Pi$  id\_presupuesto, cantidad (R4)

$\leftarrow \Pi$  sum (cantidad) (R5)

- Nombre de los clientes que han solicitado presupuestos superiores a 60.000\$.

R1  $\leftarrow \sigma$  total\_presupuesto  $\geq$  6000 (presupuesto)

R2  $\leftarrow$  R1 **JOINnat** cliente

$\leftarrow \Pi$  Nombre (R2)

# Código en SQL

```
CREATE TABLE compañía.material(  
    codigo_material varchar(30),  
    nombre varchar(30),  
    descripcion varchar(30),  
    zona varchar(30),  
    nicho varchar(30),  
    primary key(codigo_material)  
);
```

```
CREATE TABLE compañía.proveedor(  
    rut_p varchar(30),  
    nombre varchar(30),  
    direccion varchar(30),  
    telefono varchar(30),  
    primary key(rut_p)  
);
```

```
CREATE TABLE compañía.pedido_proveedor(  
    id_pedido varchar(30),  
    rut_p varchar(30),  
    fecha_pedido varchar(30),  
    total_pedido integer,  
    foreign key(rut_p) references Compañía.proveedor(rut_p),  
    primary key(id_pedido)  
);
```

```
CREATE TABLE compañía.albaran(  
    id_albaran varchar(30),  
    id_pedido varchar(30),  
    fecha_entrega varchar(30),  
    total integer,  
    foreign key(id_pedido) references Compañía.pedido_proveedor(id_pedido),  
    primary key(id_albaran)  
);
```

```
CREATE TABLE compañía.pedido_prov_material(  
    id_pedido varchar(30),  
    codigo_material varchar(30),  
    cantidad integer,  
    precio integer,  
    foreign key(id_pedido) references Compañía.pedido_proveedor(id_pedido),  
    foreign key(codigo_material) references Compañía.material(codigo_material)  
);
```

```
CREATE TABLE compania.albaran_material(  
    id_albaran varchar(30),  
    codigo_material varchar(30),  
    cantidad integer,  
    precio integer,  
    foreign key(id_albaran) references Compania.albaran(id_albaran),  
    foreign key(codigo_material) references Compania.material(codigo_material)  
);
```

```
CREATE TABLE compania.cliente(  
    rut_c varchar(30),  
    nombre varchar(30),  
    direccion varchar(30),  
    telefono varchar(30),  
    codigo_postal varchar(30),  
    primary key(rut_c)  
);
```

```
CREATE TABLE compania.presupuesto(  
    id_presupuesto varchar(30),  
    rut_c varchar(30),  
    total_presupuesto integer,  
    rebaja varchar(30),  
    fecha_presupuesto varchar(30),  
    foreign key(rut_c) references Compania.cliente(rut_c),  
    primary key(id_presupuesto)  
);
```

```
CREATE TABLE compania.pedido_cliente(  
    codigo varchar(30),  
    id_presupuesto varchar(30),  
    total_pedido_c integer,  
    fecha_pedido_c varchar(30),  
    pagado bit,  
    facturado bit,  
    foreign key(id_presupuesto) references  
Compania.presupuesto(id_presupuesto),  
    primary key(codigo)  
);
```

```
CREATE TABLE compania.pres_incluye_mat(  
    codigo_material varchar(30),  
    id_presupuesto varchar(30),  
    cantidad integer,  
    precio integer,  
    foreign key(codigo_material) references Compania.material(codigo_material),  
    foreign key(id_presupuesto) references Compania.presupuesto(id_presupuesto)  
);
```



# Conclusión

Es así como modelar un problema adecuadamente usando el MER y luego elaborando su Modelo Relacional hace que pasarlo al lenguaje SQL sea algo simple.

De esta manera pude cargar mi modelo a la plataforma PostgreSQL la cual me permitió probarlo.