

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

Se le llama base de datos a los bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés database management system o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de estos DBMS, así como su utilización y administración, se estudian dentro del ámbito de la informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas; También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

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

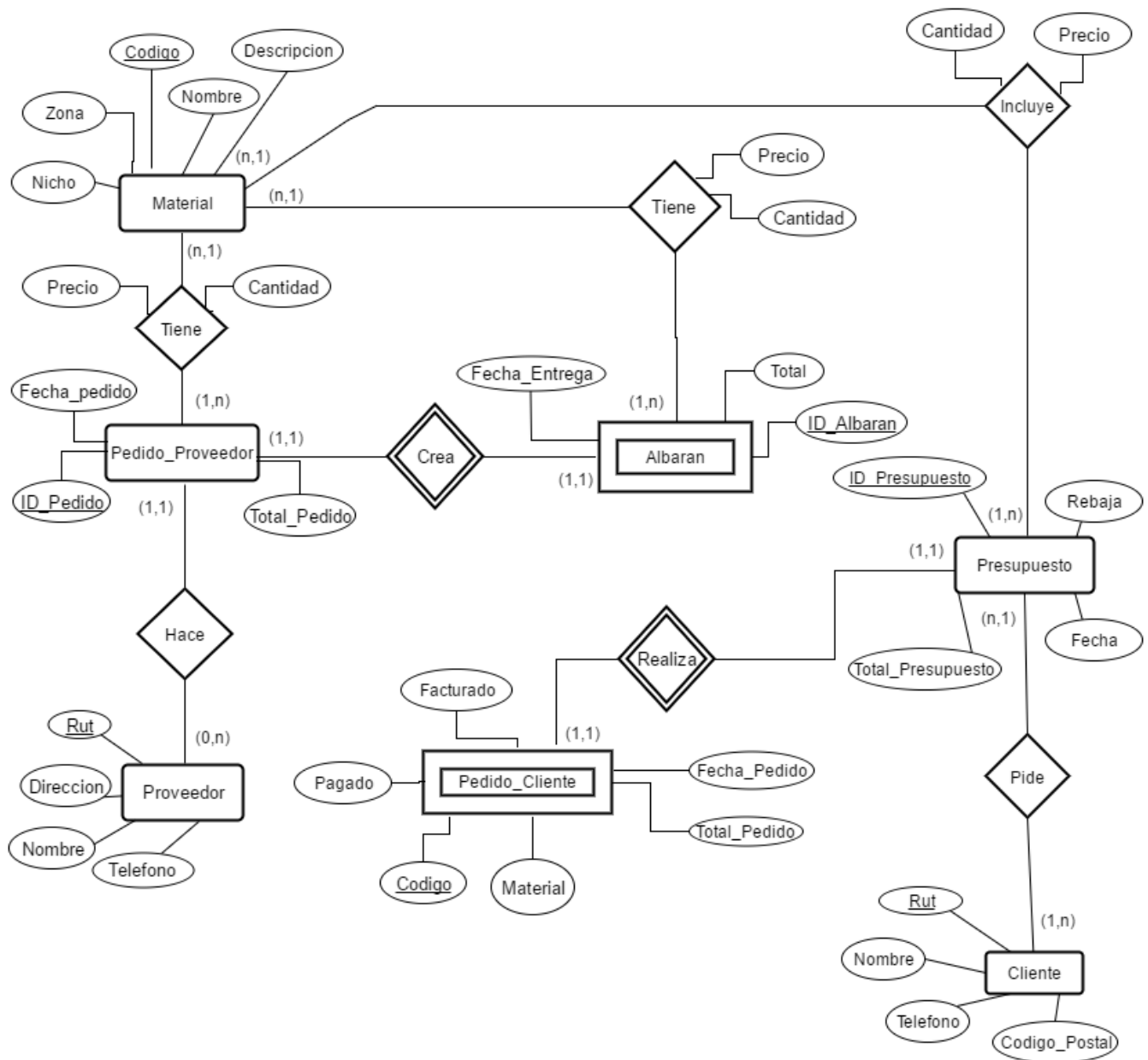
Una compañía de materias de construcción quiere hacer que su sistema de ventas será autónomo. Para aquello, deciden contratar sus servicios y, tras las reuniones iniciales, se consiguen los primeros requisitos del sistema.

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 R5 **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

R4 $\leftarrow \sigma$ fecha_pedido_c $>$ F Actual - 1 mes (R3)

$\leftarrow \Pi$ sum (cantidad) (R4)

- 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)  
);
```


Consultas en SQL

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

```
SELECT nombre
FROM proveedor
WHERE rut_p in (SELECT rut_p
                  FROM pedido_proveedor
                  WHERE fecha_pedido >= 01/01/2000 and id_pedido in(SELECT id_pedido
                                                                      FROM pedido_prov_material
                                                                      WHERE codigo_material in(
                                                                          SELECT codigo_material
                                                                          FROM material
                                                                          WHERE nombre = "hormigon
armado"))));
```

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

```
SELECT sum(cantidad)
FROM pedido_cliente
WHERE fecha_pedido_c > factual - 1mes and id_presupuesto in(SELECT id_presupuesto, cantidad
                                                             FROM pres_incluye_mat
                                                             WHERE codigo_material in(
                                                                 SELECT codigo_material
                                                                 FROM material
                                                                 WHERE nombre = "pavimento de
gres")));
```

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

```
SELECT nombre
FROM cliente
WHERE rut_c in( SELECT rut_c
                 FROM presupuesto
                 WHERE total_presupuesto >=60000);
```

Conclusión

Utilizar los Modelo Entidad Relación y el Modelo Relacional para modela un problema en particular simplifica bastante las cosas, es así como se puede tener una visión de en este caso la base de una empresa.

De esta forma llevar el modelo al lenguaje SQL se simplifica bastante para así luego ingresar los datos de la empresa y poder realizar preguntas de estos a la base de datos.