**UNIVERSIDAD DE CONCEPCIÓN**

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INFORMATÍCA 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**

**Diagrama.png**

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 🡨 pedido\_proveedor **JOINnat** pedido\_prov\_material

R2 🡨 σ Nombre = 'hormigón armado' (material)

R3 🡨 R1 **JOINnat** R2

R4 🡨 σ Fecha >= 01/01/2000 (R3)

R5 🡨 Π rut\_p (R4)

R6 🡨 R5 **JOINnat** Proveedor

🡨 Π Nombre (R6)

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

R1 🡨 pres\_incluye\_mat **JOINnat** pedido\_cliente

R2 🡨 R1 **JOINnat** material

R4 🡨 σ fecha\_pedido\_c > F Actual - 1 mes (R3)

🡨 Π sum (cantidad) (R4)

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

R1 🡨 σ total\_presupuesto >= 6000 (presupuesto)

R2 🡨 R1 **JOINnat** cliente

🡨 Π Nombre (R2)

**Código en SQL**

CREATE TABLE compañia.material(

codigo\_material varchar(30),

nombre varchar(30),

descripcion varchar(30),

zona varchar(30),

nicho varchar(30),

primary key(codigo\_material)

);

CREATE TABLE compañia.proveedor(

rut\_p varchar(30),

nombre varchar(30),

direccion varchar(30),

telefono varchar(30),

primary key(rut\_p)

);

CREATE TABLE compañia.pedido\_proveedor(

id\_pedido varchar(30),

rut\_p varchar(30),

fecha\_pedido varchar(30),

total\_pedido integer,

foreign key(rut\_p) references Compañia.proveedor(rut\_p),

primary key(id\_pedido)

);

CREATE TABLE compañia.albaran(

id\_albaran varchar(30),

id\_pedido varchar(30),

fecha\_entrega varchar(30),

total integer,

foreign key(id\_pedido) references Compañia.pedido\_proveedor(id\_pedido),

primary key(id\_albaran)

);

CREATE TABLE compañia.pedido\_prov\_material(

id\_pedido varchar(30),

codigo\_material varchar(30),

cantidad integer,

precio integer,

foreign key(id\_pedido) references Compañia.pedido\_proveedor(id\_pedido),

foreign key(codigo\_material) references Compañia.material(codigo\_material)

);

CREATE TABLE compañia.albaran\_material(

id\_albaran varchar(30),

codigo\_material varchar(30),

cantidad integer,

precio integer,

foreign key(id\_albaran) references Compañia.albaran(id\_albaran),

foreign key(codigo\_material) references Compañia.material(codigo\_material)

);

CREATE TABLE compañia.cliente(

rut\_c varchar(30),

nombre varchar(30),

direccion varchar(30),

telefono varchar(30),

codigo\_postal varchar(30),

primary key(rut\_c)

);

CREATE TABLE compañia.presupuesto(

id\_presupuesto varchar(30),

rut\_c varchar(30),

total\_presupuesto integer,

rebaja varchar(30),

fecha\_presupuesto varchar(30),

foreign key(rut\_c) references Compañia.cliente(rut\_c),

primary key(id\_presupuesto)

);

CREATE TABLE compañia.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 Compañia.presupuesto(id\_presupuesto),

primary key(codigo)

);

CREATE TABLE compañia.pres\_incluye\_mat(

codigo\_material varchar(30),

id\_presupuesto varchar(30),

cantidad integer,

precio integer,

foreign key(codigo\_material) references Compañia.material(codigo\_material),

foreign key(id\_presupuesto) references Compañia.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.