

**Ficheros y Bases de Datos**  
**Curso 2010-11**  
**Ingeniería Técnica de Informática**  
**Primer Parcial. 28-Junio-2011**

**Nombre:** \_\_\_\_\_

**Se debe entregar esta hoja**

**(4 puntos) A partir de la información sobre la BD que se describe más abajo, se pide:**

Se desea diseñar una base de datos que permita al departamento de contabilidad gestionar las ventas al por mayor de una cadena de tiendas con las siguientes características:

Para poder comprar en estas tiendas, es necesario registrarse personalmente. Una vez registrado se le asigna un código de cliente. Además de este código, se almacenará también sus datos personales, dirección de envío, datos bancarios. Un cliente puede no haber realizado ninguna compra.

Las ventas podrán realizarse también on-line, a través de su portal. Además de lo anterior, para poder llevar a cabo un compra on-line el cliente debe estar registrado en el site, tras lo cual se le asignará además un usuario y contraseña, que quedan también almacenados en su registro.

Cada tienda está caracterizada por un identificador único, la dirección física de la misma, número de empleados, volumen de facturación actual y nombre del director de la tienda. Se entiende que hay un momento en el que una tienda no ha realizado ninguna venta.

Cada tienda tiene asignado un número variable de dependientes, de los que se registra su DNI, Apellidos, Nombre y fecha de alta en la empresa. Los dependientes pueden trabajar en las distintas tiendas dependiendo de las necesidades del negocio, sin embargo por política de la empresa no pueden estar asignados a más de cuatro tiendas al mismo tiempo.

También será necesario conocer en todo momento el estado del almacén de productos para asegurarse que existe disponibilidad de mercancía.

Cada producto del almacén se identifica por un código único, descripción del producto, nº de unidades existentes en almacén, pasillo y sección del almacén donde encontrarlo. Todo producto es suministrado siempre por un proveedor y lo hace en exclusiva.

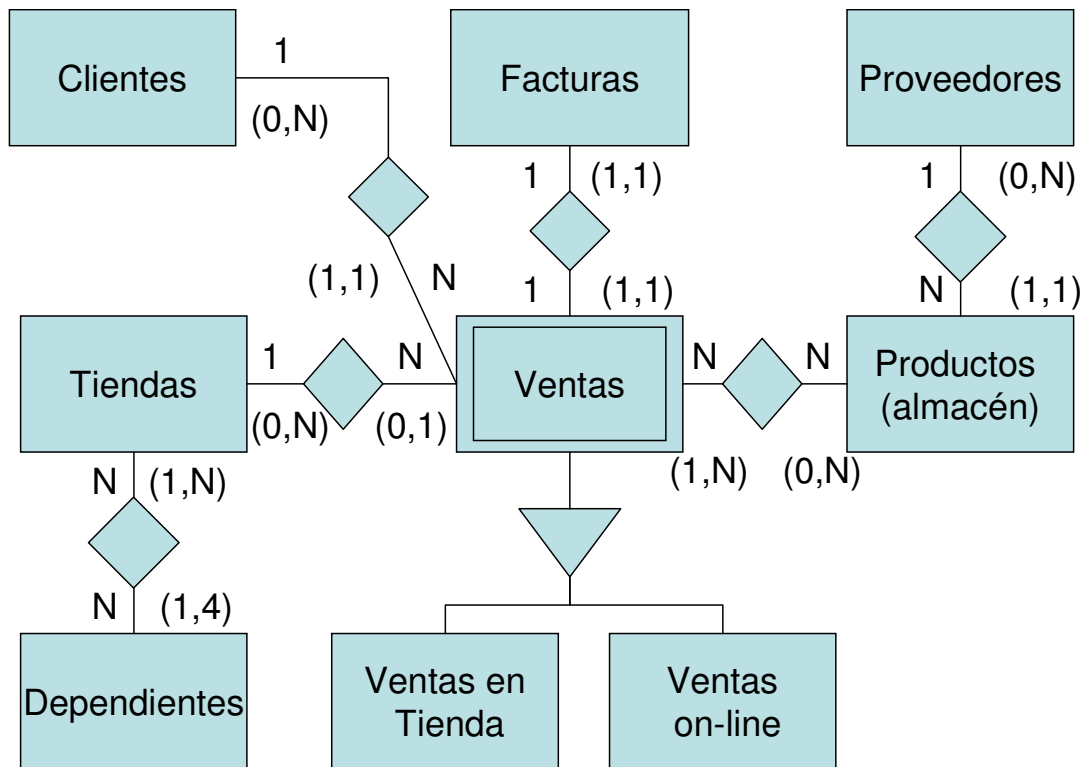
La gestión de proveedores se realizará mediante el registro de un identificador único del proveedor, datos del proveedor, teléfono y persona de contacto. Lógicamente, un proveedor suministra múltiples productos. Existen proveedores registrados que en ocasiones no suministran productos.

El jefe de contabilidad deberá poder saber siempre si la venta se realizó de una manera u otra.. Cada venta está siempre vinculada una factura, a un listado de productos y a un cliente, y puede estar asociada o no a una tienda.

Las facturas de todas las ventas son registradas con un código único, fecha, datos del cliente, descripción, importe sin IVA e importe total.

a) (2 puntos) El esquema entidad/relación o EER, incluyendo atributos clave, restricciones de cardinalidad y participación.

La entidad Ventas se puede entender como una generalización de las entidades "Ventas en Tienda" y "Ventas On-line".



Por simplificación del diagrama E-R se relacionan a continuación los atributos identificados para cada Entidad:

**Clientes**= {CodCliente, DatosPersonales, DirecciónEnvío, DatosBanco, Usuario\*, Contraseña\*}

**Tiendas** = {IDTienda, Dirección, NumEmpleados, Facturación, Director}

**Dependientes** = {DNI, Apellidos, Nombre, FechaAlta}

**Producto** = {CodProducto, DescripciónProd, Unidades, Pasillo, Sección}

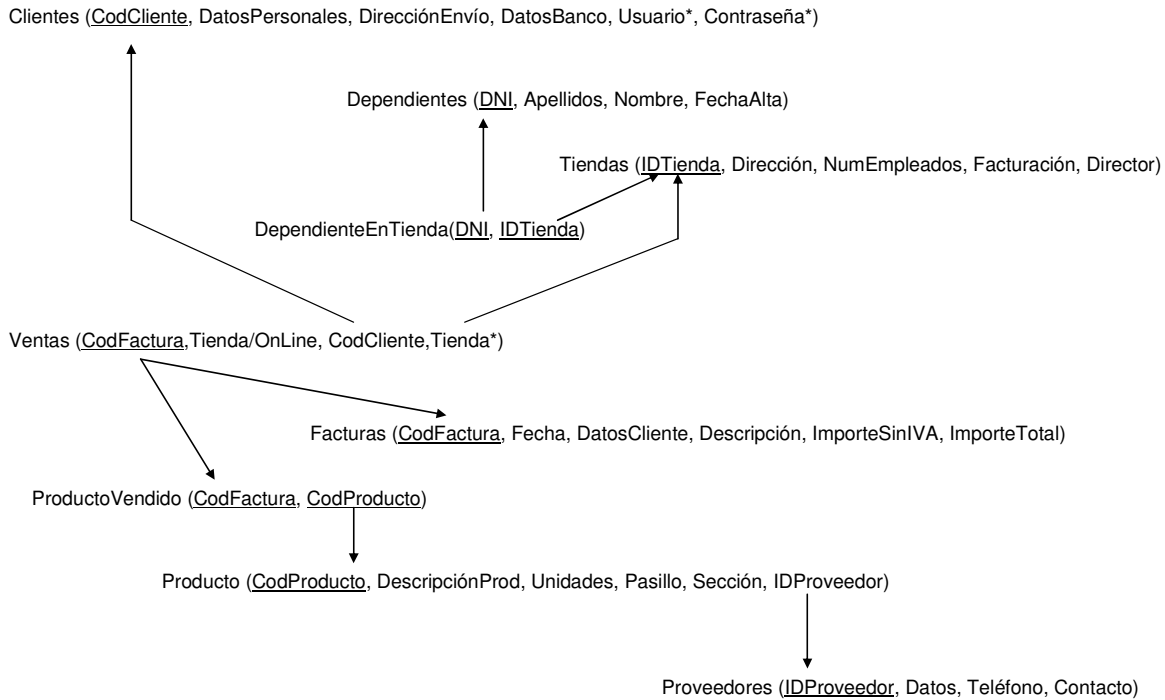
**Proveedores**= {IDProveedor, Datos, Teléfono, Contacto}

**Entidad Débil Ventas**= {Tienda/OnLine}

**Facturas**= {CodFactura, Fecha, DatosCliente, Descripción, ImporteSinIVA, ImporteTotal}

b) (1 puntos) Pasar al modelo relacional, haciendo uso de las transformaciones apropiadas.

c) (0,5 puntos) Indicar restricciones de integridad referencial resultantes.



d) (0,5 puntos) Indicar otras restricciones de integridad no reflejadas en el modelo relacional.

**De participación parcial:**

Los dependientes no pueden estar asignados a más de cuatro tiendas al mismo tiempo.

**De participación total:**

Todo producto es siempre suministrado por un proveedor y lo hace en exclusiva.

Cada venta está siempre vinculada a un listado (no vacío) de productos.

2) (3 puntos) Se pretende hacer un inventario de desastres naturales urbanos y sus damnificados, para lo cual se ha diseñado el siguiente modelo relacional:

**Desastres** (Id, tipo, magnitud, fecha, ciudad)

**Damnificados** (DNI, IdDesastre, nombre, apellido, teléfono, IdDirección, cuantía\*)

**NOTA:** Los atributos marcados con (\*) pueden tomar valores nulos.

**Realizar las siguientes consultas SQL:**

a) (0,5 puntos) Listar sin duplicados el nombre y apellido de los damnificados en 2011 que todavía no conocen alguna cuantía de subvención y su apellido empiece por 'A'. Ordenar por nombre.

```
SELECT DISTINCT nombre, apellido
FROM Damnificados, Desastres
WHERE
  IdDesastre = Id
  AND YEAR(fecha) = 2011
  AND cuantía IS NULL
  AND apellido LIKE 'A%'
ORDER BY nombre
```

b) (0,5 puntos) Mostrar ciudades con terremotos en 2011, con el número de terremotos y mayor magnitud sufrida.

```
SELECT ciudad, count(*) as NumTerremotos, MAX(magnitud) as MaxMagnitud
```

```

FROM Desastres
WHERE
    tipo = 'terremoto'
    AND YEAR(fecha) = 2011
GROUP BY ciudad

```

c) (0,5 puntos) Mostrar la ciudad que más desastres ha tenido

```

SELECT ciudad, COUNT(*)
FROM Desastres
GROUP BY ciudad
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*)
    FROM Desastres
    GROUP BY ciudad
)

```

d) (0,5 puntos) Mostrar sin duplicados el nombre y apellido de los damnificados cuyo IdDesastre no aparezca en la tabla de Desastres o que hayan sufrido más de un desastre .

```

SELECT nombre, apellido
FROM Damnificados D
WHERE NOT EXISTS (
    SELECT *
    FROM Desastres
    WHERE Id = D.IdDesastre
)
UNION
SELECT nombre, apellido
FROM Damnificados, Desastres
WHERE
    IdDesastre = Id
GROUP BY nombre, apellido
HAVING COUNT(*) > 1.

```

e) (0,5 puntos) Mostrar la suma total de las cuantías conocidas de indemnizaciones, cuantía media, y número de damnificados del mayor terremoto de Lorca de 2011.

```

SELECT IdDesastre, COUNT(*) AS num, SUM(cuantía) AS suma, AVG(cuantía) AS media
FROM Damnificados, Desastres
WHERE
    IdDesastre = Id
    AND tipo = 'terremoto'
    AND ciudad= 'Lorca'
    AND YEAR(fecha) = 2011
    AND magnitud >= ALL (
        SELECT magnitud
        FROM Desastres
        WHERE
            tipo = 'terremoto'
            AND ciudad= 'Lorca'
            AND YEAR(fecha) = 2011
    )
GROUP BY IdDesastre

```

f) (0,5 puntos) Borrar los desastres anteriores al año 2000 y sus damnificados, así como los damnificados cuyo desastre no aparezca en la tabla de desastres..

```

DELETE FROM Damnificados D
WHERE
    NOT EXISTS (
        SELECT *
        FROM Desastres
        WHERE Id = D.IdDesastreA.numAlumno
    )
OR EXISTS (
    SELECT *
    FROM Desastres
    WHERE
        Id = D.IdDesastre

```

```
AND YEAR(fecha)<2000
);
```

```
DELETE FROM Damnificados
WHERE YEAR(fecha)<2000;
```

3) (3 puntos) Dadas la relación Damnificados(DNI, IdDesastre, nombre, apellido, teléfono, IdDirección, cuantía\*) del ejercicio anterior, se quiere definir las DFs necesarias para:

- Evitar que se solicite más de una indemnización por dirección y desastre.
- Sólo se pueda almacenar un teléfono por damnificado.

Responder las siguientes preguntas justificando las respuestas.

a) (0,5 puntos) Listar un conjunto de dependencias funcionales (no triviales) para el esquema Damnificados y las restricciones descritas.

Se deben añadir dependencias funcionales a la lista de restricciones, de forma que (IdDesastre, IdDirección) sea clave candidata, además también es necesario añadir una restricción para que sólo se pueda añadir un teléfono por DNI.

```
F = {
  {DNI, IdDesastre} → {nombre, apellido, teléfono, IdDirección, cuantía},
  {IdDesastre, IdDirección} → {DNI, nombre, apellido, teléfono, cuantía},
  {DNI} → {teléfono}
}
```

b) (0,5 puntos) Calcular claves candidatas.

Las dos claves candidatas son la propia clave primaria {DNI, IdDesastre} y la clave candidata {IdDesastre, IdDirección} que hemos añadido por el apartado anterior.

c) (0,5 puntos) Calcular el recubrimiento mínimo del conjunto de dependencias funcionales resultantes del apartado a)

Un conjunto mínimo de dependencias funcionales es

```
G = {
  {DNI, IdDesastre} → {IdDirección},
  {IdDesastre, IdDirección} → {DNI},
  {IdDesastre, IdDirección} → {nombre},
  {IdDesastre, IdDirección} → {apellido},
  {IdDesastre, IdDirección} → {cuantía},
  {DNI} → {teléfono}
}
```

d) (0,5 puntos) ¿Está Damnificados en segunda forma normal? ¿por qué?

No está en 2FN, pues hay atributos (teléfono) que no forman parte de una clave candidata que dependen funcionalmente, pero no completamente de la clave candidata.

Por ejemplo, {DNI} → {teléfono} demuestra que Demuestran que {DNI, IdDesastre} → {nombre, apellido, teléfono, IdDirección, cuantía} no es una DF completa.

e) (0,5 puntos) Descomponer Damnificados en 3FN.

Damnificados no está en 3FN ya que no lo está tampoco en 2FN. Aplicando los pasos 1, 2 y 3 del algoritmo de descomposición se generan los siguientes esquemas:

R1(DNI, IdDesastre, IdDirección)

R2(IdDesastre, IdDirección, DNI, nombre, apellido, cuantía)

R3(DNI, teléfono)

f) (0,5 puntos) Descomponer Damnificados en FNBC, ¿se preservan las DFs en la descomposición?

Damnificados no está en FNBC ya que {DNI} → {teléfono} viola la definición.

R1(DNI, teléfono), está en FNBC por tener únicamente 2 atributos

R2(DNI, IdDesastre, nombre, apellido, IdDirección, cuantía), está en FNBC porque todas las DF no triviales (justamente las de F menos la última) cumplen la definición de FNBC.

Todas las DFs que estaban en el conjunto F se mantienen salvo {IdDesastre, IdDirección} → {teléfono} ya que estos atributos no están en el mismo esquema.