Diseño de un esquema conceptual

Diseño de Bases de Datos

28/11/2020

Universidad del País Vasco
Facultad de Informática

Unai Berrotaran Daniel Ruskov Elur Salgueira

Descripción del modelo

Tema: camisetas.

Tenemos los datos referentes a diferentes fábricas dedicadas a la producción de camisetas.

Una **fábrica de camisetas** está identificada mediante su *NIF*. Además conocemos su *nombre, dirección, país, número de teléfono y número de empleados* que trabajan en ella.

Cada fábrica tiene uno o varios **almacenes** donde almacenan los lotes de camisetas producidos. Estos se identifican mediante un *número único* dentro de la fábrica a la que pertenecen. Conocemos el *stock* de lotes de camisetas que hay en cada almacén.

Cada fábrica fabrica **camisetas,** que pueden ser (se dividen en) **personalizadas** (ej. un pedido particular de camisetas para una empresa como uniformes) o **no personalizadas** (ej. las camisetas correspondientes a una temporada de una tienda de ropa con diseño por defecto). Generalmente las camisetas se identifican mediante un *código*. Además, conocemos el *tipo* (manga corta o larga), *color, talla, país de fabricación* (según la fábrica) y los *materiales* que componen cada prenda. También tenemos una *descripción del diseño personalizado* de cada camiseta personalizada y una *descripción del diseño de temporada* de cada camiseta no personalizada. Finalmente, algunas de las camisetas (personalizadas y/o no personalizadas) pertenecen a una **marca** de ropa, conociendo el *nombre de la marca*.

Como se ha mencionado anteriormente, los almacenes propios de las fábricas almacenan las camisetas por *lotes.*

En una fábrica o tienda trabajan, o lo que es lo mismo, se contratan diferentes **empleados**. De cada uno conocemos su *DNI, numero de la seguridad social, conjunto de nombre y apellidos, fecha de nacimiento y edad, sexo, domicilio, telefono de contacto, fecha de inicio de la labor*, puesto que ocupa y el sueldo mensual. También interesa saber el *número de horas* que trabaja y la *jornada* en la que lo hace (mañana o tarde).

Un empleado jefe dirige la labor de los empleados en una tienda o una fábrica.

Para la venta de camisetas producidas, tenemos la información de un conjunto de **tiendas** conocidas unívocamente mediante su *NIF* y sabemos los datos de *dirección*, *número de teléfono, nombre de la tienda* y *número de empleados* que trabajan en ella.

Para que la cadena de producción-venta sea realizada, cada tienda pide sus pedidos diferenciados por el *número de pedido* y la *fecha* en la que se ha realizado a diferentes almacenes. Cada pedido es único y tiene un *precio* asociado.

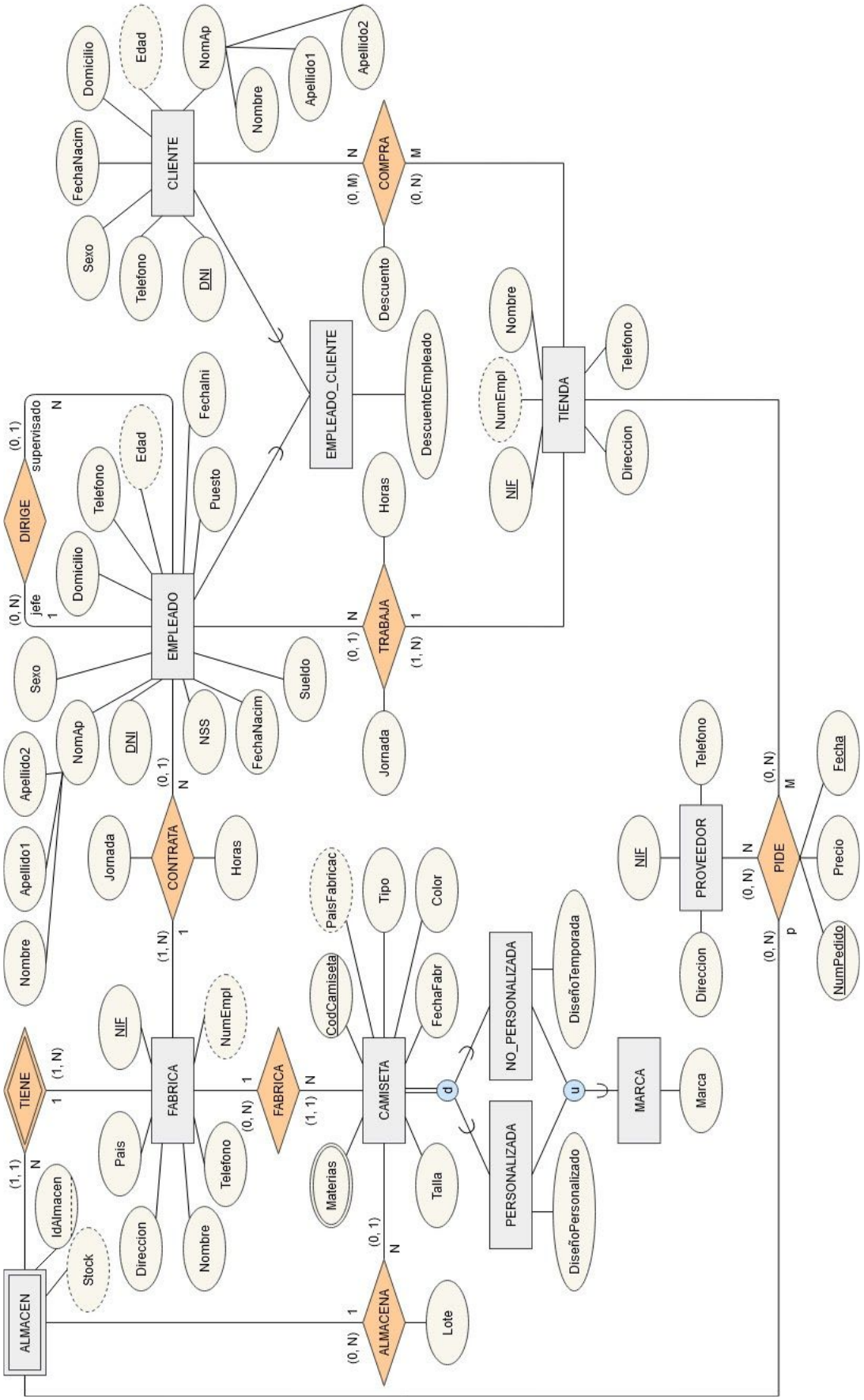
Los **proveedores** identificados mediante *NIF* y sabiendo su *dirección* y *número de teléfono*, son los encargados de recoger los pedidos de diferentes almacenes y llevarlos a las diferentes tiendas correspondientes a los que pertenecen los pedidos.

Los diferentes almacenes preparan diferentes pedidos para diferentes tiendas llevados por diferentes proveedores.

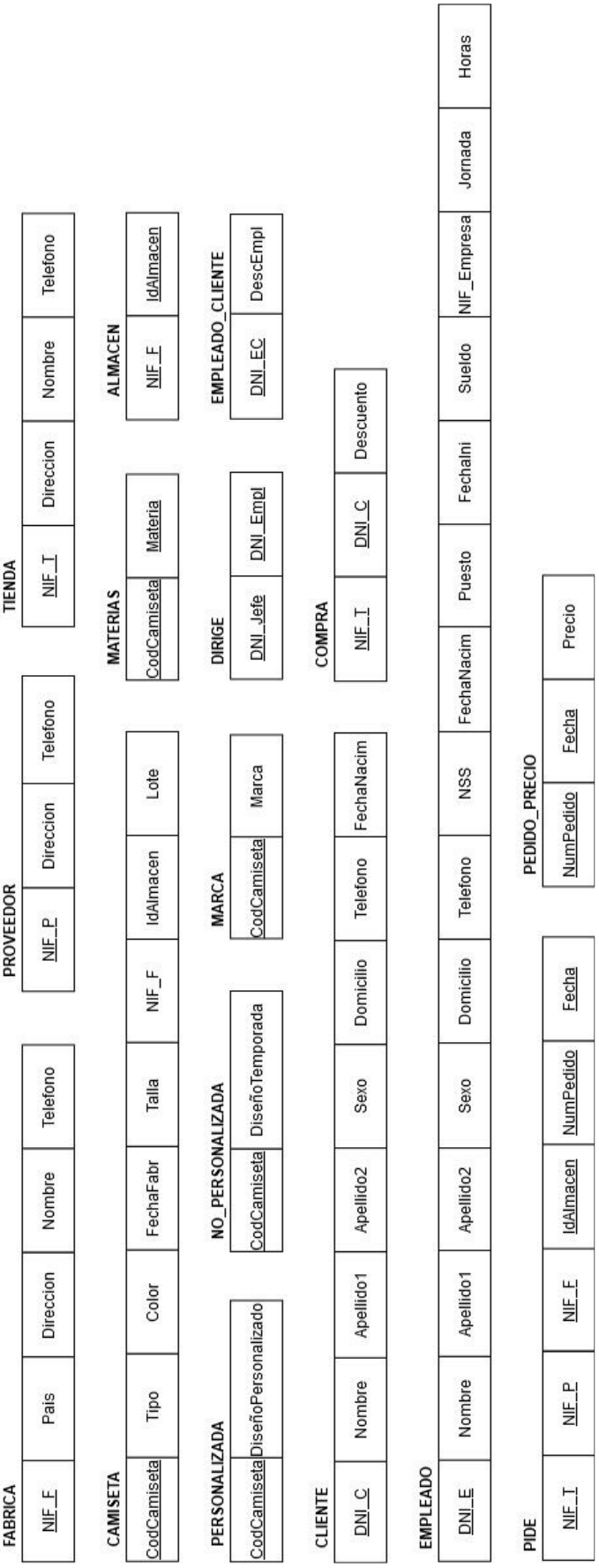
Relacionado a las tiendas, se guarda información de **clientes** que tienen una fidelidad con las tiendas. De ellos se almacenan los datos de *telefono, sexo, fecha de nacimiento, domicilio, numero de cliente, conjunto de nombre y apellidos, edad* y *DNI*, este ultimo por el cual se identifican. Cuando un cliente compra en una tienda, se le aplica un descuento por fidelidad.

Por último, están los **empleados-clientes**, que tienen un *descuento de empleado* especial y mayor al descuento de los clientes no empleados de la tienda.

Esquema E/R+


CLIENTE
Domicilio
Edad
NomAp
Nombre
Apellido1
Apellido2
FechaNacim
Sexo
Telefono
DNI
COMPRA
Descuento
EMPLEADO_CLIENTE
DescuentoEmpleado
DIRIGE
supervisado
jefe
EMPLEADO
Edad
Telefono
Domicilio
FechaIni
Puesto
Sexo
NomAp
DNI
NSS
FechaNacim
Sueldo
Apellido2
Apellido1
Nombre
TRABAJA
Horas
Jornada
TIENDA
Nombre
NumEmpl
NIF
Telefono
Direccion
CONTRATA
Jornada
Horas
TIENE
FABRICA
NIF
NumEmpl
Pais
Direccion
Nombre
Telefono
FABRICA
CAMISETA
PaisFabricac
CodCamiseta
Materias
Tipo
Color
FechaFabr
Talla
d
NO_PERSONALIZADA
PERSONALIZADA
DiseñoTemporada
DiseñoPersonalizado
u
MARCA
Marca
NIF
PROVEEDOR
Telefono
Direccion
PIDE
Fecha
Precio
NumPedido
p
ALMACEN
IdAlmacen
Stock
ALMACENA
Lote

Modelo relacional normalizado



Código generado en DBDC07

- Fichero .sql con las sentencias SQL para crear las tablas: https://drive.google.com/file/d/1nbt4wQfDjxBYcRRs1oZZ0n5UaApuAMNM/view?usp=sharing
- Fichero .sql con las sentencias SQL de inserción de tuplas: https://drive.google.com/file/d/19TaihlghnQZDD8hVbKwF7lV-ZuRz7ZJV/view?usp=sharing
- Fichero .sql con las sentencias SQL de creación de vistas, restricciones de integridad y triggers (mostradas también a continuacion): https://drive.google.com/file/d/1Di8KiD8ba3DY--zCEbnXEZfevMtnuVc0/view?usp=sharing

```
-- ============================================================================================
-- VISTAS
-- ============================================================================================
DROP VIEW INFO_CAMISETAS;
DROP VIEW EMPLEADO_VIEW;
DROP VIEW CAMISETAS_VIEW;
SELECT * FROM INFO_CAMISETAS;
SELECT * FROM EMPLEADO_VIEW;
SELECT * FROM CAMISETAS_VIEW;
-- ============================================
-- Vista que muestra la descripcion de diseño y marca de todas las camisetas
-- que son de marca junto con su codigo (personalizadas y no personalizadas)
CREATE VIEW INFO_CAMISETAS AS (
    SELECT *
    FROM (
            SELECT MARCA.CodCamiseta,
                PERSONALIZADA.DisenoPersonalizado AS DISENO,
                MARCA.MARCA
            FROM PERSONALIZADA
                JOIN MARCA ON MARCA.CodCamiseta = PERSONALIZADA.CodCamiseta
        )
    UNION
    (
        SELECT MARCA.CodCamiseta,
            NO_PERSONALIZADA.DisenoTemporada AS DISENO,
            MARCA.MARCA
        FROM NO_PERSONALIZADA
            JOIN MARCA ON MARCA.CodCamiseta = NO_PERSONALIZADA.CodCamiseta
    )
);
-- Vista que selecciona los atributos mas relevantes de los empleados
-- desde el punto de vista de las empresas
CREATE VIEW EMPLEADO_VIEW AS (
    SELECT DNI_E,
        NSS,
        NIF_EMPRESA,
        JORNADA,
        Sueldo
    FROM EMPLEADO
);
-- Vista que selecciona las camisetas y muestra los atributos necesarios
-- para poder localizarlas en los lotes de los almacenes
CREATE VIEW CAMISETAS_VIEW AS (
    SELECT CodCamiseta,
        Talla,
        NIF_F,
        IdAlmacen,
        Lote
```

```
    FROM CAMISETA
);
-- ================================================================================================
-- RESTRICCIONES DE INTEGRIDAD
-- ================================================================================================
ALTER TABLE CAMISETA DROP CONSTRAINT TALLA_VALIDA;
ALTER TABLE MATERIAS DROP CONSTRAINT MATERIALES;
ALTER TABLE CAMISETA DROP CONSTRAINT TIPO_CORRECTO;
ALTER TABLE CAMISETA DROP CONSTRAINT COD_INI_PERS;
ALTER TABLE CAMISETA DROP CONSTRAINT COD_INI_NO_PERS;
-- ================================================
-- Restriccion de integridad que comprueba que la talla introducida de una
-- camiseta es correcta (xs, s, m, l, xl, xxl, XS, S, M, L, XL, XXL).
ALTER TABLE CAMISETA
ADD CONSTRAINT TALLA_VALIDA CHECK (
        Talla IN (
            'xs',
            's',
            'm',
            'l',
            'xl',
            'xxl',
            'XS',
            'S',
            'M',
            'L',
            'XL',
            'XXL'
        )
    );
-- Restriccion de integridad que comprueba que los materiales introducidos
-- de una camiseta son correctos (algodon ringspun, algodon hilado,
-- algodon pre-encogido, poliester)
ALTER TABLE MATERIAS
ADD CONSTRAINT MATERIALES CHECK (
        Materia IN (
            'algodon ringspun',
            'algodon hilado',
            'algodon pre-encogido',
            'poliester'
        )
    );
-- Restriccion de integridad que comprueba que el tipo introducido de
-- camiseta es correcto (personalizada, n personalizada)
ALTER TABLE CAMISETA
ADD CONSTRAINT TIPO_CORRECTO CHECK (
        Tipo IN (
            'personalizada',
            'n personalizada'
        )
    );
-- Restriccion de integridad que verifica si una camiseta es personalizada,
-- entonces su codigo empieza por 1.
ALTER TABLE CAMISETA
ADD CONSTRAINT COD_INI_PERS CHECK (
        NOT (
            Tipo = 'personalizada'
            AND NOT (CodCamiseta LIKE '1%')
        )
    );
```

```
-- Restriccion de integridad que verifica si una camiseta es no personalizada,
-- entonces su codigo empieza por 0.
ALTER TABLE CAMISETA
ADD CONSTRAINT COD_INI_NO_PERS CHECK (
        NOT (
            Tipo = 'no personalizada'
            AND NOT (CodCamiseta LIKE '0%')
        )
    );
-- ================================================================================================
-- TRIGGERS
-- ================================================================================================
DROP TRIGGER INSERTAR_EN_SUBCLASES;
DROP TRIGGER MAX_SUELDO;
DROP TRIGGER CLIENTES_MAYORES_DE_EDAD;
-- ================================================
-- Trigger que al insertar en la tabla camisetas, automaticamente y segun el codigo de camiseta
-- inserta la camiseta en la subclase disjunta correspondiente con descripcion null para que
-- pueda ser modificado posteriormente con facilidad mediante rango de codigos. Tambien se activa
-- al modificar y eliminar (pero al tratarse de FK y PK hay RI que actuan antes)
CREATE OR REPLACE TRIGGER INSERTAR_EN_SUBCLASES
AFTER INSERT OR UPDATE OR DELETE OF CodCamiseta ON CAMISETA
FOR EACH ROW
BEGIN
    IF INSERTING THEN
        IF (:NEW.CodCamiseta LIKE '1%') THEN
            INSERT INTO PERSONALIZADA VALUES (:NEW.CodCamiseta, NULL);
        ELSE
            INSERT INTO NO_PERSONALIZADA VALUES (:NEW.CodCamiseta, NULL);
        END IF;
    END IF;
    IF UPDATING THEN
        IF ((:NEW.CodCamiseta LIKE '1%') AND (:OLD.CodCamiseta LIKE '0%')) THEN
            DELETE FROM NO_PERSONALIZADA WHERE CodCamiseta = :OLD.CodCamiseta;
            INSERT INTO PERSONALIZADA VALUES (:NEW.CodCamiseta, NULL);
        END IF;
        IF ((:NEW.CodCamiseta LIKE '0%') AND (:OLD.CodCamiseta LIKE '1%')) THEN
            DELETE FROM PERSONALIZADA WHERE CodCamiseta = :OLD.CodCamiseta;
            INSERT INTO NO_PERSONALIZADA VALUES (:NEW.CodCamiseta, NULL);
        END IF;
    END IF;
    IF DELETING THEN
        IF (:NEW.CodCamiseta LIKE '1%') THEN
            DELETE FROM PERSONALIZADA WHERE CodCamiseta = :OLD.CodCamiseta;
        ELSE
            DELETE FROM NO_PERSONALIZADA WHERE CodCamiseta = :OLD.CodCamiseta;
        END IF;
    END IF;
END;
-- Trigger que obliga que el sueldo de los empleados no supere el valor de  5000
CREATE OR REPLACE TRIGGER MAX_SUELDO
BEFORE INSERT OR UPDATE OF Sueldo ON EMPLEADO
REFERENCING NEW AS INSERTADO
FOR EACH ROW WHEN (INSERTADO.Sueldo > 5000)
BEGIN
    Raise_Application_Error (
        -20343,
        'Error, el salario maximo es de 5000 por persona'
    );
END;
```

```
-- Trigger que impide fidelizar a clientes menores de edad
CREATE OR REPLACE TRIGGER CLIENTES_MAYORES_DE_EDAD
BEFORE INSERT ON CLIENTE
FOR EACH ROW
DECLARE MENORES_EXCEPTION EXCEPTION;
BEGIN
    IF ((months_between(SYSDATE, :NEW.FechaNacim)/12) < 18) THEN
        RAISE MENORES_EXCEPTION;
    END IF;
    EXCEPTION
    WHEN MENORES_EXCEPTION THEN
        Raise_Application_Error (
        -20434,
        'Error, el cliente a registrar no puede ser menor de edad'
    );
END;
```