Diseño y creación de la Base de Datos

Fase 2

JuanPaulo

Índice

Contenido

Modelo Entidad Relación

1. Entidades Identificadas

Se añade la tabla Provincias que en la fase de análisis no se había contemplado para poder gestionar mejor las provincias a las que pertenecen los clientes

- **Clientes**: Hace referencia a los clientes
- **Empleados**: hace referencia a los empleados de la tienda
- **Provincias**: hace referencias a las provincias
- **Reparaciones**: hace referencia a las reparaciones de la tienda
- **Acciones**: hace referencia a las acciones que se realizan en las reparaciones como pueden ser: instalación de software, limpieza de virus, etc.
- **Artículos**: Hace referencia a los artículos que se venden en la tienda
- **Compra**: hace referencia a las compras que realizan los clientes
- **Lineas_detalle:** hace referencia a los detalles de la compra, conteniendo los artículos o reparaciones de la compra
- **Usuarios**: hace referencia a los usuarios del sistema

La entidad Lineas_detalle se considera una entidad débil de la entidad Factura ya que si se elimina una factura no tendría sentido que se guardaran las líneas de detalle ya que existen porque existe la factura. Se considera una entidad débil en identificación ya que para identificarse necesitaría también la clave de la entidad factura

2. Atributos de las entidades identificadas

- **Clientes**: id_cliente, DNI, nombre, apellidos, dirección, ciudad, CP, teléfono, mail
- **Empleados**: id_empleado, DNI, nombre, apellidos, dirección, provincia, ciudad, teléfono
- **Provincias**: cod_provincia, provincia
- **Reparaciones**: id_Reparacion, diagnostico, fecha_entrada, fecha_salida, observaciones,estado
- **Acciones**: id_accion, nombre, precio, descripción
- **Artículos**: id_articulo, nombre, precio, unidades, descripción
- **Compra**: NºCompra, IVA, Fecha
- **Lineas_Detalle**: Cod_Lienea, Cantidad
- **Usuarios**: Usuario, Password, Tipo

*El atributo estado de la tabla Reparaciones puede ser (reparacion,reparado,facturar,facturado).
*Reparación: está en reparación
*Reparado: ya está reparado
*Facturar: esta para facturar
*Facturado: ya está facturado

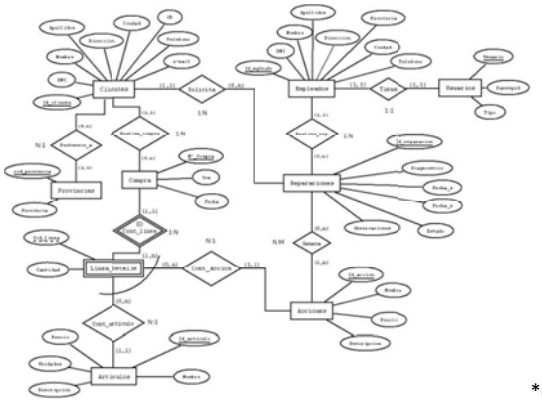
* Los tipos de usuario pueden ser administradores o empleados

3. Relaciones y cardinalidades

- **Cliente-Reparaciones**: se relacionan mediante “Solicita”. La relación es 1:N.Un Cliente puede solicitar ninguna o varias reparaciones y una reparación puede ser solicitada por un cliente.
- **Cliente-Compra**: se relacionan mediante “Realiza_compra”. La relación es 1:N.Un Cliente puede realizar ninguna o varias compras y una compra puede ser realizada por un cliente.
- **Cliente-Provincia**: Se relacionan mediante “pertenece_a”. La relación es 1:N. Un cliente puede pertenecer a una provincia y a una provincia pueden pertenecer varios clientes
- **Empleado-Reparaciones**: se relacionan mediante “Realiza_rep”. La relación es 1:N.Un Empleado puede realizar ninguna o varias reparaciones y una reparación puede ser realizada por un empleado.
- **Empleado-usuario**: se relacionan mediante “tiene”. La relación es 1:1.Un Empleado puede tener un usuario y un usuario puede pertenecer a un empleado solamente.
- **Reparaciones-Acciones**: se relacionan mediante “Genera”. La relación es N:M. Una Reparación puede generar una o varias acciones y una acción puede ser generada por ninguna o por varias reparaciones.
- **Compra-Linea_Detalle**: se relacionan mediante “Cont_Linea”. La relación es 1:N.Una Compra puede contener una o varias líneas de detalle y una Linea puede ser contenida por una compra.
- **Linea_Detalle-Acciones**: se relacionan mediante “Cont_Accion”. La relación es 1:N.Una Línea de detalle puede contener una acción y una acción puede estar en ninguna o en muchas líneas de detalle.
- Linea_Detalle-Articulos: se relacionan mediante “Cont_Articulo”. La relación es 1:N.Una Línea de detalle puede contener un artículo y un artículo puede estar en ninguna o en muchas líneas de detalle.

La relaciones “Linea_Detalle-Acciones” y “Linea_Detalle-Articulo” mantienen una relación de exclusividad porque un línea de detalle solo va a contener o una instancia de articulo o de acción pero no las 2 a la vez.

4. Diagrama E-R



*el archivo E-R.jpeg contiene la imagen del diagrama E-R

Modelo Relacional

1. Paso del modelo E-R al modelo relacional

a. Todas las entidades se convierten en una relación (Tabla)

- Clientes(**id_cliente**,dni,Nombre,Apellidos,Direccion ,Ciudad,CP,Telefono,mail)
- Empleados(**id_empleado**,dni, nombre, apellidos, dirección, provincia, ciudad, teléfono)
- Provincias(**cod_provincia**, provincia)
- Reparaciones(**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones,estado)
- Acciones(**id_accion**, nombre, precio, descripción)
- Artículos(**id_articulo**, nombre, precio, unidades, descripción)
- Compra (**NºCompra**, IVA, Fecha)
- Linea_Detalle(**Cod_Lienea,NºCompra**, Cantidad) la manera de transformar una dependencia en identificación es utilizar el mecanismo de propagación de clave. Se propaga la clave primaria de la entidad fuerte a la entidad débil pasando a formar esta parte de su clave primaria. En este caso se propaga la clave primara de Compra a Linea_detalle pasando a formar parte de su clave primaria
- Usuarios(**Usuario**, Password, Tipo)

b. Relaciones

- **N:M:** se convierten en tablas propagando las claves primarias de las tablas que asocia y los atributos de la relación.
  En este caso:

  Reparaciones-Acciones: Genera (**id_Reparacion,Id_Accion**)

- **1:1**:

  - Si la relación es entre entidades con cardinalidades (0,1) y (0,1), es mejor crear una relación para evitar tener muchos nulos como propagación de alguna de las claves a la otra.

  - Si la relación es entre entidades con cardinalidades (0,1) y (1,1), es mejor propagar la clave de la entidad (1,1) a la (0,1).

  - Si la relación es entre entidades con cardinalidades (1,1) y (1,1), la propagación es indiferente, y se hará atendiendo a los criterios de frecuencia de acceso (consulta, modificación, inserción, etc.) a cada una de las tablas en cuestión

  Empleado-usuario: en este caso se propaga la clave de la entidad Usuario a la entidad Empleado, quedando:

  Empleado (**id_empleado** ,Dni, nombre, apellidos, dirección, provincia, ciudad, teléfono, usuario)

- **1:N** Estas interrelaciones se pueden transformar de dos maneras diferentes:

    - Propagar la clave principal de la entidad que tiene cardinalidad máxima 1 a la que tiene N, y hacer desaparecer la tabla de la relación como tal. Si la relación tuviera pocos atributos asociados, estos atributos pasan a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima muchos.

    - Transformarla en una nueva tabla como si fuese de una relación de tipo N:M, es decir, incluyendo los atributos de la relación y las claves primarias de las dos entidades. Esta acción es recomendable sólo:
        - cuando es posible que aparezcan muchos nulos porque existen pocos elementos relacionados,
        - o cuando se prevé que dicha relación pasará en un futuro a ser de tipo N:M,
        - o cuando la relación tiene 2 o más atributos propios.

    Cliente-Reparaciones: en este caso se propaga la clave de la relación Cliente a la relación Reparaciones, quedando:

    - Reparaciones (**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones, estado, id_cliente)

    Empleado-Reparaciones, en este caso se propaga la clave de la relación Empleado a la relación Reparaciones, quedando:

    - Reparaciones (**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones, estado, id_cliente,id_empleado)

    Cliente-Compra: en este caso se propaga la clave de la relación Cliente a la relación Compra, quedando:

    - Compra(**NºCompra**, IVA, Fecha, id_cliente)

    Linea_Detalle-Acciones: en este caso se propaga la clave de la relación Acciones a la relación Linea_detalle, quedando:

    - Linea_Detalle(**Cod_Lienea,NºCompra**, Cantidad,Id_Accion)

    Linea_Detalle-Artículos: en este caso se propaga la clave de la relación Artículos a la relación Linea_detalle, quedando:

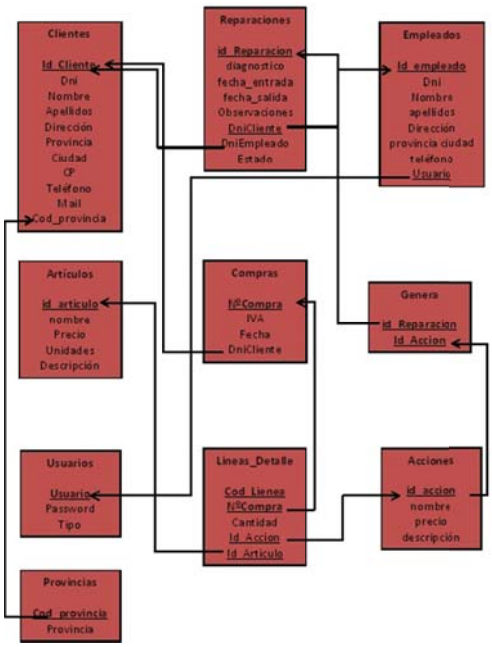
    - Linea_Detalle(**Cod_Lienea,NºCompra**, Cantidad,Id_Accion,Id_Articulo)

    Cliente-Provincias: en este caso se propaga la clave de la relación Provincias a Clientes quedando:

- Clientes(**id_cliente**,dni,Nombre,Apellidos,Direccion,Ciudad,CP,Telefono,mail, cod_provincia)

c. El modelo relacional quedara de la siguiente forma:

- Clientes(**id_cliente**, Dni, Nombre, Apellidos, Dirección, Ciudad, CP, Teléfono, mail, cod_provincia)
- Empleado(**id_empleado**,Dni, nombre, apellidos, dirección, provincia, ciudad, teléfono, usuario)
- Provincias(**Cod_provincia**,provincia)
- Reparaciones (**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones, estado , id_cliente, id_empleado)
- Acciones(**id_accion**, nombre, precio, descripción)
- Artículos(**id_articulo**, nombre, precio, unidades, descripción)
- Compras(**NºCompra**, IVA, Fecha, id_cliente)
- Linea_Detalle(**Cod_Lienea, NºCompra**, Cantidad, Id_Accion,Id_Articulo)
- Genera (**id_Reparacion,Id_Accion**)
- Usuarios(**Usuario**, Password, Tipo)



*el archivo ModeloR.jpg contiene la imagen del m
relacional

Normalización

Clientes (**id_cliente**,Dni, Nombre, Apellidos, Dirección, Ciudad, CP, Teléfono, mail, Cod_provincia)
Empleado (**id_empleado**,Dni, nombre, apellidos, dirección, provincia, ciudad, teléfono, usuario)
Provincias (**Cod_provincia**,provincia)
Reparaciones (**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones, estado , id_liente,id_empleado)
Acciones (**id_accion**, nombre, precio, descripción)
Artículos (**id_articulo**, nombre, precio, unidades, descripción)
Compras(**NºCompra**, IVA, Fecha, id_cliente)
Linea_Detalle (**Cod_Lienea, NºCompra**, Cantidad, Id_Accion,Id_Articulo)
Genera (**id_Reparacion,Id_Accion**)
Usuarios (**Usuario**, Password, Tipo)

1. Primera forma normal

- Se ve claramente que todos los atributos de las tablas son atómicos por lo tanto están en 1FN

- Dependencias Funcionales

    - Tabla Clientes
        - Id_empleado(Dni,Nombre, Apellidos, Dirección, Provincia,Ciudad, CP, Teléfono, mail
        - CP(Ciudad
    - Tabla Empleados
        - Id_empleado(Dni, nombre, apellidos, dirección, provincia, ciudad, teléfono, usuario
    - Tabla Reparaciones
        - id_Reparacion(diagnostico, fecha_entrada, fecha_salida, observaciones, estado , Id_empleado, Id_empleado
    - Tabla Acciones
        - id_accion(nombre, precio, descripción
    - Tabla Articulo
        - id_articulo(nombre, precio, unidades, descripción
    - Tabla Compras
        - NºCompra (IVA, Fecha, Id_empleado
    - Linea_Detalle
        - Cod_Lienea, NºCompra(Cantidad, Id_Accion,Id_Articulo
    - Usuarios
        - Usuario(Password, Tipo

2. Segunda Forma Normal

Las tablas están en 2FN ya que están en 1FN y todos los atributos no clave dependen por completo de la clave primaria.

3. Tercera Forma Normal

La tabla clientes es la única que tiene una dependencia funcional transitiva ya que a través del código postal se puede averiguar la ciudad .
Como ya hay una tabla que hace referencia a las provincias se decide no crear una tabla para el código postal y las ciudades quedando todas las tablas normalizadas.

4. Tablas resultantes del proceso de normalización

- Clientes (**Id_cliente**,Dni, Nombre, Apellidos, Dirección, CP, Teléfono, mail, Cod_provincia)
- Provincias (**Cod_provincia**,provincia)
- Empleado (**id_empleado**,Dni, nombre, apellidos, dirección, provincia, ciudad, teléfono, usuario)
- Reparaciones (**id_Reparacion**, diagnostico, fecha_entrada, fecha_salida, observaciones, estado, Id_cliente, Id_empleado)
- Acciones (**id_accion**, nombre, precio, descripción)
- Artículos (**id_articulo**, nombre, precio, unidades, descripción)
- Compras(**NºCompra**, IVA, Fecha, Id_cliente)
- Linea_Detalle (**Cod_Lienea, NºCompra**, Cantidad, Id_Accion,Id_Articulo)
- Genera (**id_Reparacion,Id_Accion**)
- Usuarios (**Usuario**, Password, Tipo)

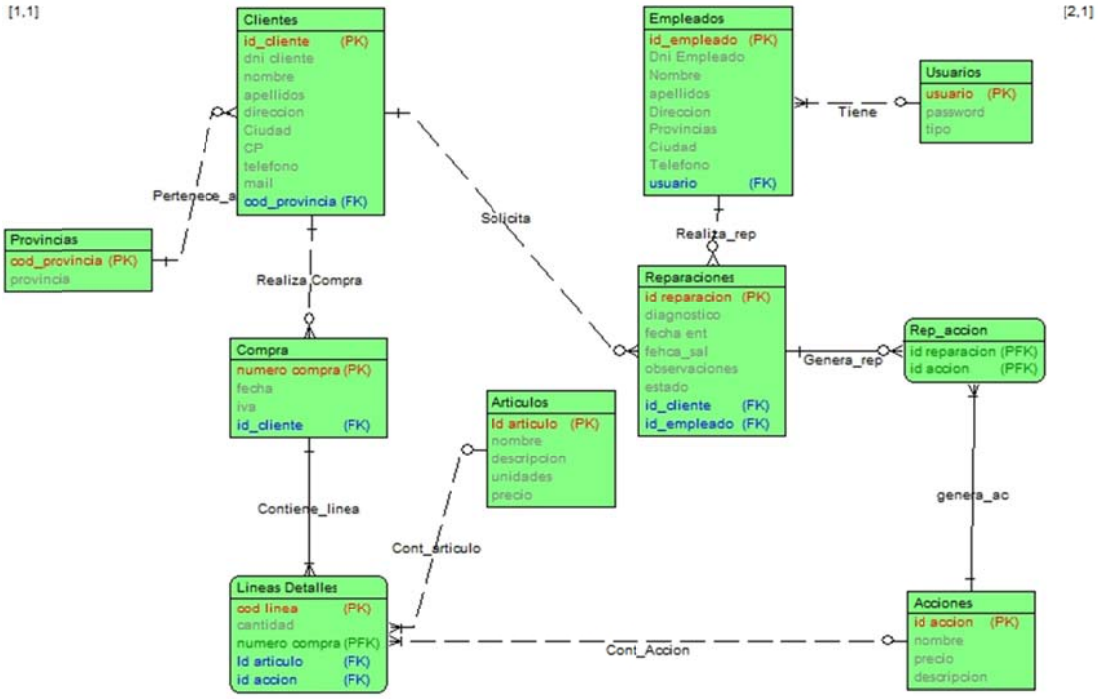
Diseño de la base de datos con herramienta CASE

Herramientas utilizadas en la fase de diseños y creación de la base de datos

- Toad Data modeler V. 2.29 para el diseño
- MySQL-5.1.4 como gestor de base de datos

1. Modelo relacional generado con Toad Data modeler

En Entregas\FaseII\Fase2_Toad se encuentra el fichero .dm2



2. Información de la BD, Scripts generados y pruebas realizadas

Triggers de la Base de datos

Se han añadido a la base de datos los siguientes triggers:

- fecha_compra : Este trigger se ejecutara cuando se inserte una compra en la BD y servirá para añadir la fecha actual del sistema en el campo “fecha” de la tabla Compra

- fecha_rep_ent: Este trigger se ejecutara cuando se inserte una reparación en la BD y servirá para añadir la fecha actual del sistema en el campo “fecha_ent” de la tabla Reparaciones

- fecha_rep_sal: Este trigger se ejecutara cuando se cambie el estado de una reparación a “reparado” en la BD y servirá para añadir la fecha actual del sistema en el campo “fecha_sal” de la tabla Reparaciones

- fecha_rep_sal: Este trigger se ejecutara cuando se cambie el estado de una reparación a “reparado” en la BD y servirá para añadir la fecha actual del sistema en el campo “fecha_sal” de la tabla Reparaciones

- modifica_stock: Este trigger se ejecutara cuando se inserte una línea de detalles con un artículo en la BD y servirá para actualizar el stock del artículo que se añadió en las líneas de detalles de las compras.

Vistas de la Base de datos

Se han añadido las siguientes vistas:

- facturas artículos: listara las compras con los artículos de los clientes
- facturas reparaciones: listara las compras con las reparaciones de los clientes

Scripts Generados

Se generaron los siguientes scripts:

- Tienda.sql: Este script contiene toda la base de datos con los registros incluidos
- BD.sql : Script que solo contiene la creación de la BD sin los registros
- inserta_datos.sql: solo contiene los registros de la BD

Para cargar la BD se puede hacer de 2 formas:

- ejecutando el script Tienda.sql (este genera toda la BD incluido los datos)
- o cargar el script BD.sql (que genera la BD) y el script inserta_datos.sql(que inserta los datos en la BD)

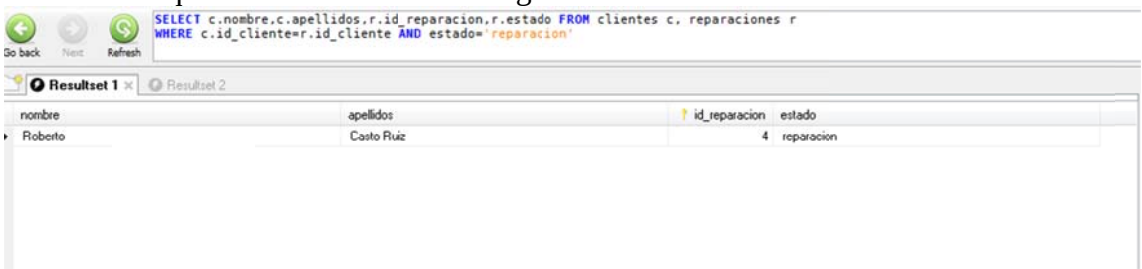
Consultas realizadas

Se realizan las siguientes consultas:

- Listado de clientes que tienen un equipo en reparación:

  SELECT c.nombre,c.apellidos,r.id_reparacion,r.estado FROM clientes c, reparaciones r
  WHERE c.id_cliente=r.id_cliente AND estado='reparacion'

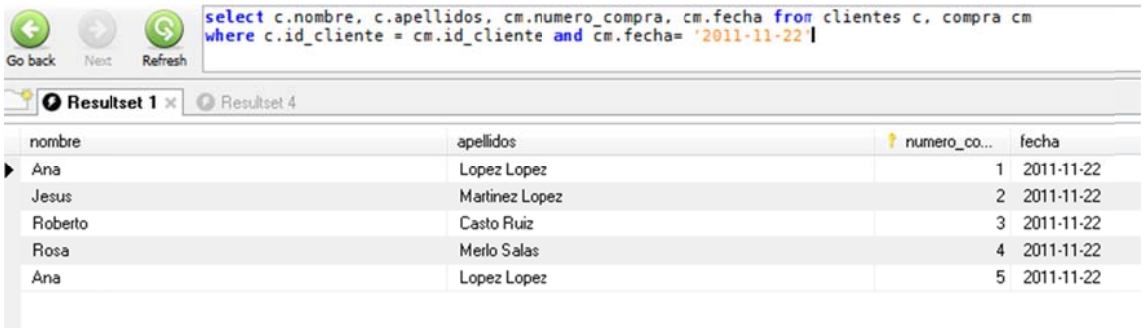
  El resultado que se debe obtener es el siguiente:



- Listado de clientes que compraron en una fecha determinada

  select c.nombre, c.apellidos, cm.numero_compra, cm.fecha from clientes c, compra cm
  where c.id_cliente = cm.id_cliente and cm.fecha= '2011-11-22'

  El resultado que se debe obtener es el siguiente:


Go back
Next
Refresh
select c.nombre, c.apellidos, cm.numero_compra, cm.fecha from clientes c, compra cm
where c.id_cliente = cm.id_cliente and cm.fecha= '2011-11-22'
Resultset 1
Resultset 4
nombre
apellidos
numero_co...
fecha
Ana
Lopez Lopez
1
2011-11-22
Jesus
Martinez Lopez
2
2011-11-22
Roberto
Casto Ruiz
3
2011-11-22
Rosa
Merlo Salas
4
2011-11-22
Ana
Lopez Lopez
5
2011-11-22