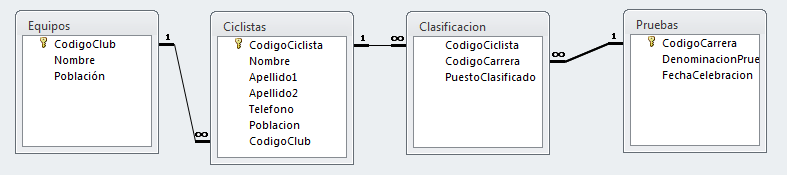
*Se desea hacer una base de datos para llevar el control de una competición deportiva, se trata de un circuito de bicicletas, de manera que se disputan una serie de carreras al año, en las que participan o no los distintos ciclistas que pertenecen a distintos clubes. Después de que nuestro analista haya estudiado el problema nos propone el siguiente diseño:*



*Ten en cuenta las siguientes consideraciones:*

* *Los campos que son claves primarias (y foráneas) son de tipo entero.*
* *El campo ‘FechaCelebración’ es de tipo fecha.*
* *El resto de campos son de tipo texto.*
* *Para todos los campos es obligatorio que se cumplimenten, salvo el de CodigoClub de la tabla ciclista, ya que las normas de la competición permiten participar a ciclistas que no pertenezcan a ningún club.*
* *Para todas las modificaciones de las claves primaria que se actualicen sus referencias en cascada. Para ello vamos a crear restricciones en las tablas que correspondan los nombres serán los siguientes EC (Tabla Ciclista-Equipo), CC (Ciclistas-Clasificacion) y CP (Clasificacion-Pruebas).*
* *Deseamos que se cumpla la Integridad Referencial por lo que deseamos que el ‘motor’ de todas las tablas sea INNOBD.*

**1.-** En un fichero de texto denominado ciclismo.sql, crea las sentencias sql para crear la base de datos de ciclismo y las tablas descritas. Ten en cuenta además del tipo de dato de cada campo los que son clave primaria y foránea en cada caso (2,5 puntos).

**ARCHIVO:** ciclismo.sql

drop database ciclismo;

create database ciclismo;

use ciclismo;

create table Equipos(

CodigoClub Integer PRIMARY KEY,

Nombre VARCHAR(20) NOT NULL,

Poblacion VARCHAR(20) NOT NULL

)ENGINE=INNODB;

create table Ciclistas (

CodigoCiclista Integer PRIMARY KEY,

Nombre VARCHAR(20) NOT NULL,

Apellido1 VARCHAR(20) NOT NULL,

Apellido2 VARCHAR(20) NOT NULL,

Poblacion VARCHAR(20) NOT NULL,

CodigoClub Integer,

CONSTRAINT CE FOREIGN KEY(CodigoClub) REFERENCES Equipos(CodigoClub) ON DELETE SET NULL ON UPDATE CASCADE

)ENGINE=INNODB;

create table Pruebas (

CodigoCarrera Integer PRIMARY KEY,

DenominacionPrueba VARCHAR(20) NOT NULL,

FechaCelebracion DATE NOT NULL

)ENGINE=INNODB;

create table Clasificacion(

CodigoCiclista Integer NOT NULL,

CodigoCarrera Integer NOT NULL,

PuestoClasificado Integer NOT NULL,

CONSTRAINT CC FOREIGN KEY (CodigoCiclista) REFERENCES Ciclistas(CodigoCiclista) ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT CP FOREIGN KEY (CodigoCarrera) REFERENCES Pruebas(CodigoCarrera) ON DELETE CASCADE ON UPDATE CASCADE,

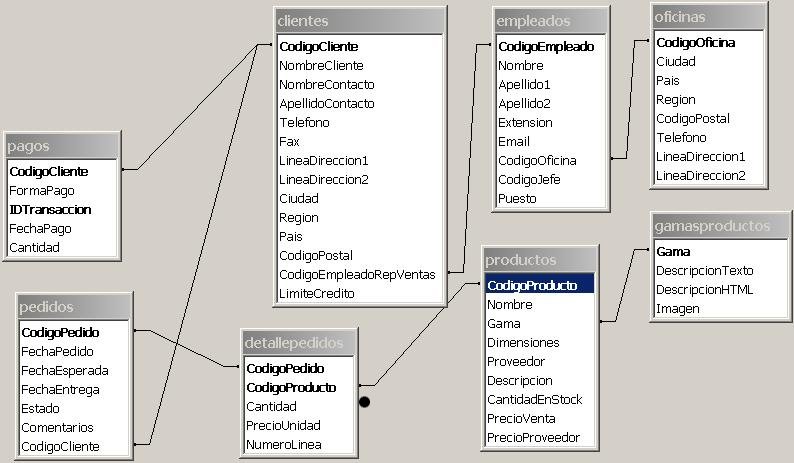
CONSTRAINT PK PRIMARY KEY (CodigoCiclista,CodigoCarrera)

)ENGINE=INNODB;

**2.-** Después de crear las tablas nos hemos dado cuenta de que hay algunos detalles que no hemos tenido en cuenta por lo que debemos crear un fichero de texto llamado modificaciones.sql que va a contener las sentencias SQL necesarias para solucionar las siguientes incidencias (1,5 punto):

1. En la tabla Ciclistas es necesario añadir un nuevo campo obligatorio denominado ‘Sexo’ que podrá tener como valores permitidos: ‘Hombre’ o ‘Mujer’.
2. Después del campo ‘Apellido2’ añade un campo de tipo entero corto que se llame ‘Edad’.
3. Renombra la tabla ‘Equipos’ para que se llame ‘Clubes’;
4. Elimina la tabla Clubes.
5. Elimina el campo ‘CodigoClub’ de la tabla Ciclistas.
6. **ARCHIVO:** Modificaciones.
7. # a.
8. ALTER TABLE Ciclistas ADD Sexo ENUM('Hombre','Mujer') NOT NULL;
9. # b.
10. ALTER TABLE Ciclistas ADD Edad TINYINT AFTER Apellido2;
11. # c.
12. RENAME TABLE Equipos TO Clubes;
13. # d.
14. ALTER TABLE Ciclistas DROP FOREIGN KEY CE;
15. DROP TABLE Clubes;
16. # e.
17. ALTER TABLE Ciclistas DROP CodigoClub;

**3.-** Usando la base de datos de jardinería en MySQL realizar las siguientes consultas.



1. Visualizar todos los campos de la tabla pagos y además el nombre del cliente que lo ha realizado. (0,5 puntos)
2. Obtener el número de clientes que tiene cada país ordenados por número de clientes en orden descendente. (0,5 puntos)
3. Listar el nombre del cliente, el nombre de su representante y el del jefe del representante de cada cliente ordenado por el Apellido1 y Nombre del Jefe del representante. (0,5 puntos)
4. Obtener el nombre del producto y el precio del producto más barato. (0,5 puntos)
5. Sacar un listado de los 15 códigos de productos más pedidos ordenado por cantidad pedida. (0,5 puntos)
6. Sacar el nombre, apellido, oficina (ciudad) y cargo del empleado que no representa a ningún cliente. (0,5 puntos)
7. Listar todos los datos de los empleados y sus clientes asociados, tienen que aparecer los empleados sin cliente asociado. (0,5 puntos)
8. Sacar un listado con el precio sin IVA “base imponible” y precio con IVA “precio total” de cada pedido. (0,5 puntos)
9. Obtener el pedido más caro del empleado que menos cliente tiene. (1 punto)

# 3 a.

select Pagos.\*,Clientes.NombreCliente from Pagos natural join Clientes;

# 3 b.

select Pais, count(\*) as 'Numero de Clientes' from Clientes group by Pais order by count(\*) Desc;

# 3 c.

Select NombreCliente, e.Nombre,e.Apellido1,j.Nombre,j.Apellido1

from Empleados as e inner join Empleados as j on e.CodigoJefe=j.CodigoEmpleado

inner join Clientes on e.CodigoEmpleado=CodigoEmpleadoRepVentas

order by j.Apellido1,j.Nombre;

# 3 d.

select Nombre, PrecioVenta from Productos where PrecioVenta=(select min(PrecioVenta) from Productos);

# 3 e.

Select Nombre, sum(cantidad) as 'Total Pedido'

from DetallePedidos natural join Productos

group by Nombre

Order by sum(cantidad) desc

limit 15;

# 3 f.

Select CodigoEmpleado, Nombre, Apellido1, Apellido2, Ciudad from Empleados natural join Oficinas where CodigoEmpleado not in (Select CodigoEmpleadoRepVentas from Clientes) Order by CodigoEmpleado;

# 3 g.

Select Empleados.\*,Clientes.\* from Empleados left join Clientes on CodigoEmpleado=CodigoEmpleadoRepVentas;

# 3 h.

Select CodigoPedido,sum(Cantidad\*PrecioUnidad) as 'Base Imponible', sum(cantidad\*PrecioUnidad)\*1.18 as 'Precio Total'

from DetallePedidos

group by CodigoPedido;

#i

**Los pedidos de cada cliente.**

Select CodigoPedido, CodigoCliente, sum(Cantidad\*PrecioUnidad) as total from Pedidos natural join DetallePedidos group by CodigoPedido,CodigoCliente;

**Lista los empleados que menos Clientes tienen**

Select CodigoEmpleadoRepVentas from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcli from Clientes group by CodigoEmpleadoRepVentas having numcli= (Select min(numcliente) from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcliente from Clientes group by CodigoEmpleadoRepVentas) as totalcli)) as CliMin;

**Listar los clientes de los empleados que menos clientes tienen**

Select CodigoCliente from Clientes where CodigoEmpleadoRepVentas in (Select CodigoEmpleadoRepVentas from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcli from Clientes group by CodigoEmpleadoRepVentas having numcli= (Select min(numcliente) from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcliente from Clientes group by CodigoEmpleadoRepVentas) as totalcli)) as CliMin);

**Listar los pedidos de los clientes de los empleados que menos clientes tienen**

Select CodigoPedido, CodigoCliente, sum(Cantidad\*PrecioUnidad) as total from Pedidos natural join DetallePedidos group by CodigoPedido,CodigoCliente in (Select CodigoCliente from Clientes where CodigoEmpleadoRepVentas in (Select CodigoEmpleadoRepVentas from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcli from Clientes group by CodigoEmpleadoRepVentas having numcli= (Select min(numcliente) from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcliente from Clientes group by CodigoEmpleadoRepVentas) as totalcli)) as CliMin));

Ahora sacaría el máximo de ellos

Select max(total) from (Select CodigoPedido, CodigoCliente, sum(Cantidad\*PrecioUnidad) as total from Pedidos natural join DetallePedidos group by CodigoPedido,CodigoCliente in (Select CodigoCliente from Clientes where CodigoEmpleadoRepVentas in (Select CodigoEmpleadoRepVentas from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcli from Clientes group by CodigoEmpleadoRepVentas having numcli= (Select min(numcliente) from (Select CodigoEmpleadoRepVentas,count(CodigoCliente) as numcliente from Clientes group by CodigoEmpleadoRepVentas) as totalcli)) as CliMin))) as final;