

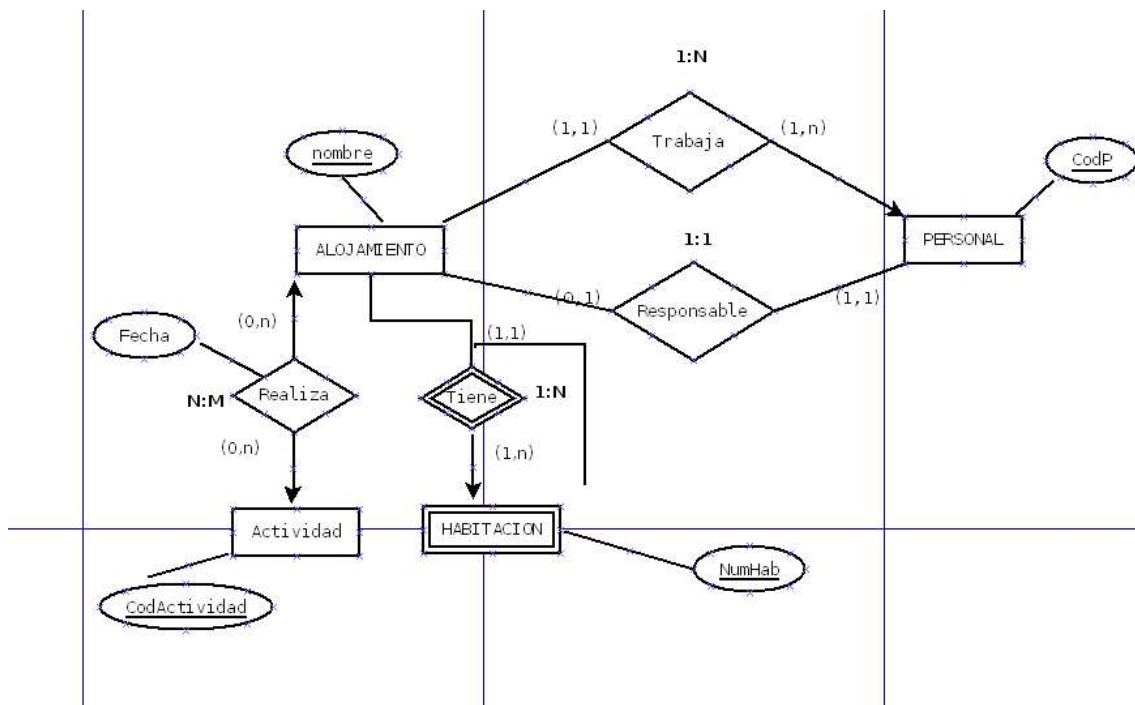
ALOJAMIENTOS RURALES

La comunidad de Madrid desea guardar información sobre los alojamientos rurales que existen en dicha comunidad. Para ello decide crear una BBDD que recoja las siguientes consideraciones:

Un alojamiento rural se identifica por un nombre ("Tío Pepe", "Las Rosas", etc.) tiene una dirección, un teléfono y una persona de contacto que pertenece al personal del alojamiento. En cada alojamiento trabajan una serie de personas que se identifican por un código de personal. Se requiere conocer el nombre completo, la dirección y el NIF del personal. Aunque en un alojamiento trabajen varias personas, una persona solo puede trabajar en un alojamiento. Los alojamientos se alquilan por habitaciones y se desea conocer cuántas habitaciones componen el alojamiento, de qué tipo es cada una de estas habitaciones (individuales, dobles, triples), si poseen cuarto de baño y el precio. Cada habitación tiene un número distinto.

En algunos de estos alojamientos se realizan actividades multiaventura organizadas para huéspedes (senderismo, bicicleta de montaña, etc.). Estas actividades se identifican por un código. Se quiere saber el nombre de la actividad, la descripción y el nivel de dificultad de dicha actividad (de 1 a 10). Estas actividades se realizan un día a la semana, por ejemplo: en la casa "Las Rosas" se practica senderismo los jueves y se desea guardar esta información. Pero puede haber algún día en el que no se realice ninguna actividad.

Diseñar el Modelo Conceptual y el Relacional.



ALOJAMIENTOS (nombreA, direccion, telefono, contacto*(FK), numHabitaciones)

PERSONAL (codigoP, nombreA (FK), NIF, nombreP, dir)

Entre ALOJAMIENTOS y PERSONAL existe un bucle referencial que impide la creación de las mismas esto se solucionará creando la tabla ALOJAMIENTOS y una vez creadas las dos tablas se modificará la tabla ALOJAMIENTOS para añadir contacto como una clave foránea de CódigoP en PERSONAL.

HABITACIONES (nombreA (FK), codH, tipo, banio, precio)

ACTIVIDADES (codActividad, nivel, nombreActividad, descripcion)

REALIZA_ACTIVIDAD (codActividad (FK), nombreA (FK), diaSemana)

Contacto se considera optativo para posibilitar que cuando se da de alta un alojamiento, la persona de contacto todavía no se haya asignado (NULL).

Si día no se considera parte de la clave primaria, no se podría realizar la misma actividad en un alojamiento en varios días

Considerándola como clave primaria, se permite realizar la misma actividad en un alojamiento pero no en un mismo día.

Mediante un trigger se controlará que un alojamiento trabaje al menos una persona. También se controlará que la persona de contacto trabaje en el alojamiento.

```
Create table ALOJAMIENTOS (  
    nombreA Varchar(20) NOT NULL,  
    direccion Varchar(20) NOT NULL,  
    telefono Varchar(20) NOT NULL,  
    contacto Char(3),  
    numHabitaciones Int NOT NULL,  
    Primary Key (nombreA));  
  
Create table PERSONAL (  
    codigoP Char(3) NOT NULL,  
    nombreA Varchar(20) NOT NULL,  
    NIF Char(9) NOT NULL,  
    nombreP Varchar(20) NOT NULL,  
    dir Varchar(20) NOT NULL,  
    UNIQUE (NIF),  
    Primary Key (codigoP),  
    Foreign Key (nombreA) references ALOJAMIENTOS (nombreA) on delete cascade on  
update cascade);  
  
Alter table ALOJAMIENTOS  
    add Foreign Key (contacto) references PERSONAL (codigoP) on delete set  
NULL on update cascade;  
  
Create table HABITACION (  
    codH Char(3) NOT NULL,  
    tipo Varchar(20) NOT NULL,  
    banio Boolean NOT NULL,  
    precio Float NOT NULL,  
    nombreA Varchar(20) NOT NULL,  
    Primary Key (nombreA, codH),  
    Foreign Key (nombreA) references ALOJAMIENTOS (nombreA) on delete cascade  
on update cascade);  
  
Create table ACTIVIDAD (  
    codActividad Char(3) NOT NULL,  
    nivel Int NOT NULL,  
    nombreActividad Varchar(20) NOT NULL,
```

```
        descripcion Varchar(50) NOT NULL,  
        UNIQUE (nombreActividad),  
        Primary Key (codActividad),  
        CHECK (nivel BETWEEN 1 and 10));  
  
Create table REALIZA_ACTIVIDAD (  
        nombreA Varchar(20) NOT NULL,  
        codActividad Char(3) NOT NULL,  
        diaSemana Varchar (10) NOT NULL,  
        Primary Key (nombreA,codActividad),  
        Foreign Key (codActividad) references ACTIVIDAD (codActividad)  
        on delete cascade on update cascade,  
        Foreign Key (nombreA) references ALOJAMIENTOS (nombreA)  
        on delete cascade on update cascade);
```