

Realizar un diagrama Entidad-Relación del siguiente problema presentado:

Se desea diseñar un juego de estrategia en tiempo real por internet que reúna los siguientes requisitos.

Los jugadores crearán cuentas de juego. De los jugadores no interesa guardar nada, pues sus datos estarán en la propia cuenta de juego.

De las cuentas necesitamos guardar su ID que será único para cada cuenta, el Nick (nombre de usuario) que también será único y el email del usuario, único también. Además la cuenta también tendrá una contraseña. Otros datos a guardar de la cuenta serán la fecha de creación, puntos totales, cantidad de barcos mercantes y la cantidad de oro almacenada.

Cada cuenta tiene ciudades. De cada ciudad interesa guardar su ID que será único, su nombre y nivel. Además, una ciudad almacena una determinada cantidad de madera, piedra y carne, que serán materiales que se usarán para un cometido concreto. Una cuenta puede tener varias ciudades y cada ciudad solo podrá pertenecer a una sola cuenta.

Cada ciudad tendrá una serie de edificios de los cuales se guardará su código, nombre y nivel actual. Dicho edificio solo pertenece a una ciudad concreta y las ciudades pueden tener varios edificios.

Las ciudades se ubican en islas. Una ciudad solo puede estar en una isla, en cambio en la isla puede haber varias ciudades. El límite de ciudades en cada isla es un valor que se concretará en el desarrollo de la aplicación dependiendo de la tecnología usada. Por el momento diremos que tiene muchas.

Cada isla tendrá un ID propio, un nombre y unas coordenadas en el eje X y unas coordenadas en el eje Y. Por ejemplo, una isla con coordenadas 50,50 estará situada en el centro del mapa.

Las cuentas pueden crear clanes. De cada clan guardamos su ID, nombre, abreviatura y fecha de creación. Una cuenta puede crear muchos clanes, pero cuando crea uno, será miembro de ese clan y no podrá crear otro hasta que deje ese clan o lo borre. Un clan es creado por una sola cuenta.

El clan, una vez creado, tendrá como miembro a su creador y luego puede tener muchos más miembros, que serán las cuentas de juego. Por lo tanto, un clan puede tener varias cuentas y una cuenta solo puede pertenecer a un clan.

Además, un clan puede formar alianzas con otros clanes, donde cada clan puede tener alianzas con más de un clan y ser aliado de uno o más clanes.

Las cuentas se pueden enviar mensajes entre ellas. De los mensajes queremos guardar un código único, el texto del mensaje y la fecha en la que fue enviado. La cuenta al enviar un mensaje se convierte en emisor y la cuenta que recibe el mensaje, en receptor. Una cuenta puede enviar y/o recibir varios mensajes, pero cada mensaje solo tendrá un receptor y un emisor.

Los jugadores se pueden atacar entre sí usando ejércitos. En esta primera versión, los ejércitos estarán asociados a cada ciudad y solo podrán moverse para atacar a otras ciudades de la misma isla. Por lo tanto, cada ciudad tendrá un ejército propio. De cada ejército queremos saber solamente un código propio y el número de soldados del que está compuesto. Un ejército solo puede pertenecer a una ciudad y una ciudad solo tendrá un ejército.

Cuando un jugador quiere atacar a otro, se genera una batalla por lo que dicha batalla estará relacionada con una ciudad como atacante y otra ciudad como defensora. Como cada ejército está ligado a cada ciudad, no será necesario indicar en la batalla que ejércitos van a participar, pero si que será necesario guardar un código de batalla, una fecha, la cantidad de bajas que existen en cada bando (bajas defensoras, bajas atacantes), además de las materias robadas en dicha batalla (madera, piedra y carne). Una ciudad puede participar en varias batallas, tanto como atacante como defensora, pero en una batalla concreta solo tendrá una ciudad atacante y otra ciudad defensora.

1. Crear un diseño entidad relación con todas las entidades, relaciones, atributos clave y cardinalidades pertinentes. (6 puntos)
2. Pasar el modelo entidad relación a modelo relacional. (2,5 puntos)
3. Crear dicha base de datos en MySQL. (1,5 puntos)

Ejemplo de base de datos:

```
CREATE DATABASE IF NOT EXISTS mibasededatos;
use mibasededatos;
create table if not exists tabla1(
    codigo int auto_increment unique not null,
    campo1 varchar(45) no null,
    campo2 date,
    campo3 bigint,
    primary key (codigo));

create table if not exists tabla2(
    codigo int auto_increment unique not null,
    codigotabla1 int not null,
    campo1 varchar(45) not null,
    campo2 date,
    foreign key (codigotabla1) references tabla1(codigo) on delete no action,
    primary key (código, codigotabla1));
```