

BOLSA DE TRABAJO - TÉCNICOS INFORMÁTICOS DE APLICACIONES

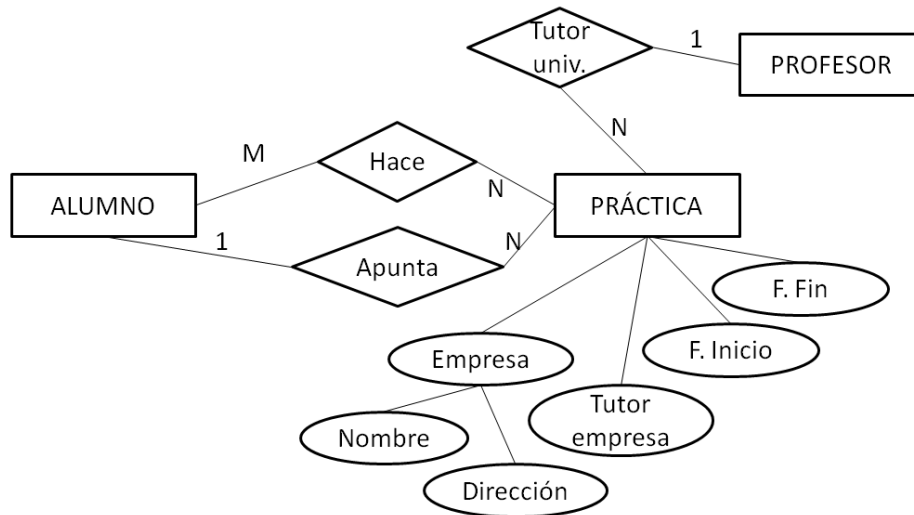
EJERCICIO PRÁCTICO 1

La universidad ha decidido desarrollar un sistema propio de asignación de carga docente de los profesores que imparten clase, esto es, del reparto de clases que van a dar. Para ello nos han proporcionado la siguiente descripción del sistema y de los datos que se desean guardar.

- La universidad está organizada en centros, en los que se imparten dos tipos de titulaciones (grado y postgrado).
- Cada centro tiene un identificador, el nombre que es único, y la dirección (calle, código postal, ciudad, provincia). En cada centro se pueden impartir una o varias titulaciones, aunque también existen centros en los que no se imparte docencia.
- Por cada titulación, independientemente del tipo, se desea guardar el nombre y un código, compuesto por una letra que indica el grupo al que pertenece (ciencias, ingeniería, arquitectura, humanidades,...) y una cifra de tres dígitos, que hará que el código sea único para cada titulación. Cada titulación no es exclusiva de cada centro, lo que permite impartir una titulación en varios centros.
- Las asignaturas, que pertenecen a una única titulación, tienen un código que las identifica, nombre, número de créditos, así como el tipo de docencia que tiene (magistral, grupo de aula, grupo de ordenador o seminario), siendo uno o varios los tipos de docencia que tenga una asignatura.
- Por cada asignatura, la universidad decide cuantos grupos se van a dar dependiendo de la previsión de los alumnos que se van a matricular. Por cada grupo se guardará un código que consta de dos letras que indican si el grupo tiene docencia de mañana (MA) o de tarde (TA) seguido de un número (01, 02, 03...) dependiendo del número de grupos que haya de la asignatura a la que corresponde el grupo. Además, también se guardará el número de clases magistrales, grupos de aula, grupos de ordenador y seminarios que tendrá. Para cada grupo habrá un profesor como responsable, el cual tendrá que ser doctor. Aunque varios podrán ser los profesores los que impartan clase en él, sin importar si son o no doctores, siendo importante guardar el número de créditos que imparte cada uno en cada grupo.
- Los profesores, de los cuales se guardarán el DNI, el nombre, el primer apellido, el segundo, la dirección de correo electrónico y el teléfono, tienen que pertenecer a un único centro por cuestiones administrativas, pero pueden impartir clase en varios centros, donde en cada uno de ellos tendrá asignado un despacho y se almacenará el código de despacho asignado. Los profesores pueden ser de dos tipos, doctores y no doctores. En el caso de los doctores se desea guardar desde cuando lo son, y tendrán además la posibilidad de ser coordinadores de una titulación, no siendo posible asumir varias coordinaciones a la vez ni compartir la coordinación con otro compañero. Hay que tener constancia de desde cuando ostenta el cargo cada coordinador.

Se pide:

1. El diagrama entidad-relación de la base de datos y las tablas correspondientes al diagrama realizado.
2. Con la misma base de datos se quiere hacer una aplicación que gestione las prácticas de empresa de los alumnos y alguien ha diseñado la siguiente solución:



El procedimiento de la gestión de empresas es el siguiente:

- La empresa se registra en el sistema
- El responsable del centro valida la empresa
- La empresa introduce las prácticas
- Los alumnos seleccionan las prácticas que desean
- La empresa escoge un alumno de los que han seleccionado la práctica

Discuta si es correcta o no la solución y expón los motivos.

3. Dado el siguiente script SQL, haga un estudio de ingeniería inversa para deducir cuál es el esquema conceptual de la base de datos subyacente a la que se hace referencia, ignorando cualquier instrucción que no defina elementos de la base de datos.

```
create domain t_dni as adecimal(8,0);
create domain t_age as integer check (value between 16 and 99);
create table person as
  (DNI      t_dni,
   nExp     char(8) not null,
   name     varchar(25) not null,
   surname  varchar(25) not null,
   age     t_age,
   primary key (DNI),
   unique (nExp));
create table work as
  (code     char(8),
   name     varchar(80) not null,
   primary key (code));
```

```

create table teacher as
  (DNIteacher t_dni,
   email      varchar(50)
   primary key(DNIteacher),
   unique(email),
   foreign key (DNIteacher) references person (DNI)
     on delete cascade on update cascade);
create table student as
  (DNItstudent t_dni,
   age         t_age,
   primary key (DNItstudent),
  foreign key (DNItstudent) references person (DNI)
     on delete set null on update cascade);
create table publish as
  (code       char(8),
   DNI        t_dni,
   course     varchar(9),
   primary key (code, DNI, curso),
   foreign key (code) references work (code)
     on delete cascade on update cascade,
   foreign key (DNI) references teacher (DNIteacher)
     on delete cascade on update cascade);
create table supervise as
  (code       char(8),
   DNIt       t_dni,
   DNIs       t_dni,
   primary key (code, DNIt, DNIs),
   foreign key (code) references work (code)
     on delete cascade on update cascade,
   foreign key (DNIt) references teacher (DNItteacher)
     on delete cascade on update cascade)
   foreign key (DNIs) references teacher (DNItstudent)
     on delete cascade on update cascade);
create table updated_exps as
  (old_exp char(8),
   new_exp char(8),
   primary key (old_exp, new_exp));
create table newoldstudents as
  (DNI       t_dni,
   primary key (DNI));
create view A as
  (select nExp, name, surname
   from person inner join student
     on person.DNI = student.DNItstudent
   where age <25);
create index tdni on person(DNI);
create procedure P (IN A T_exp, IN B T_exp)
  begin
    Insert into updated_exps values (A, B)
  end;
create trigger T1 after insert on person statement when age > 55
  begin
    insert into newoldstudents (select DNI from new_table)
  end;
create trigger T2 before update of nexp on student for each row
  begin
    call P(old.nexp, new.nexp)
  end;

```