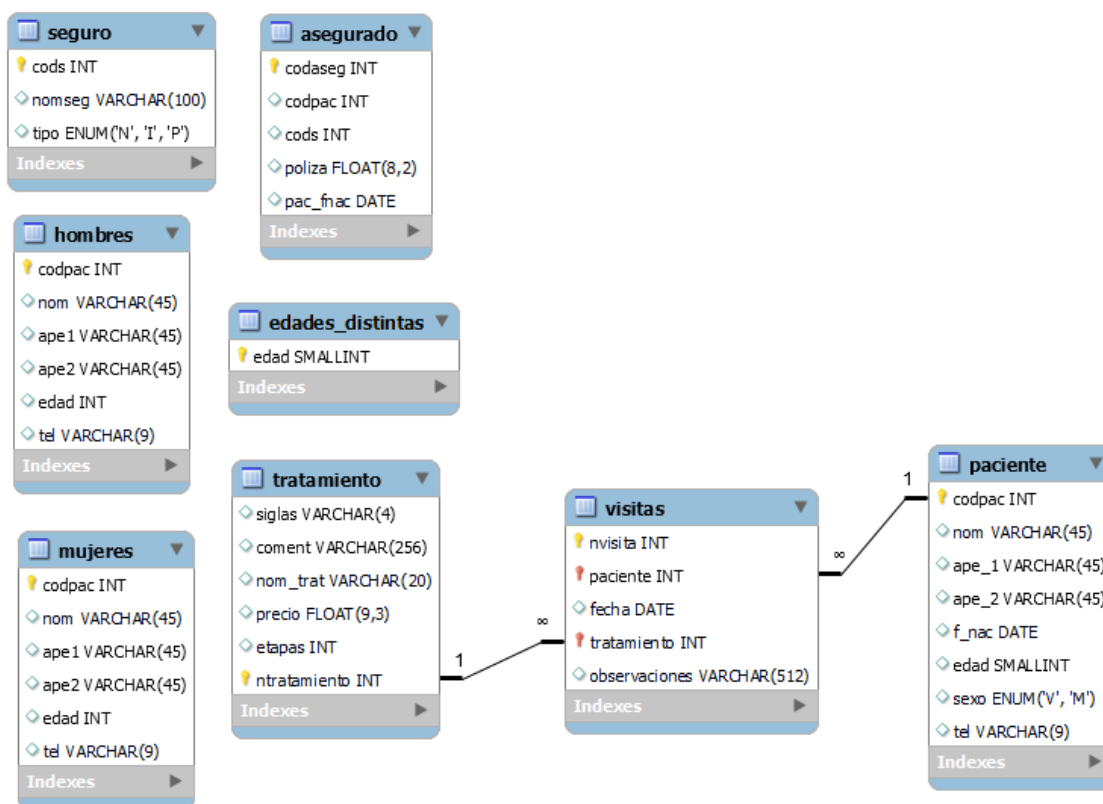


### Ejercicio 7.1.

Dentro de la BBDD dentista, implementad los siguientes *triggers*:

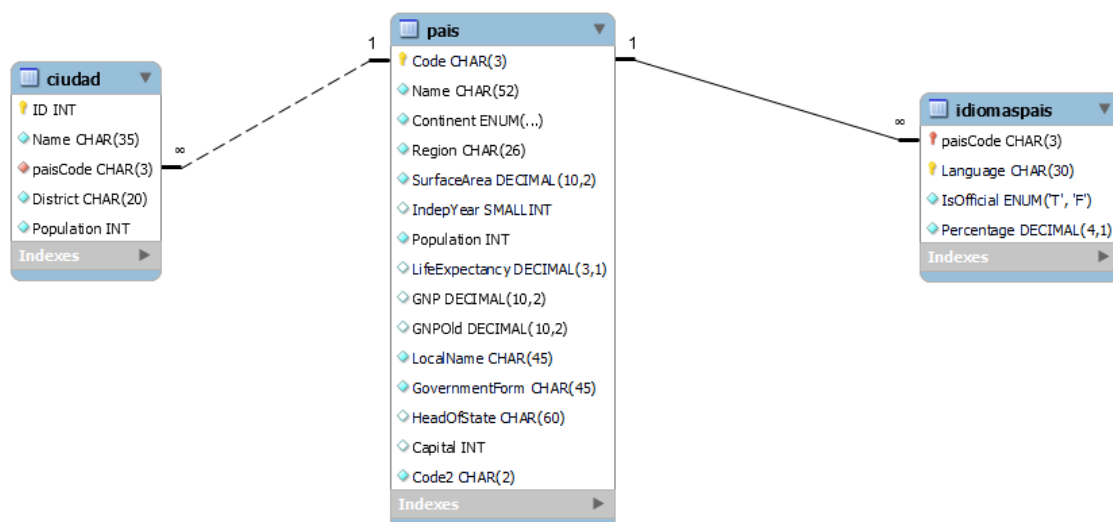


1. Crea un TRIGGER que se encargue de, ANTES de que se produzca una inserción de un paciente, ACTUALICE su edad a la edad real del día actual.
2. Crea un TRIGGER que se encargue de, DESPUÉS de que se produzca la inserción de un paciente, inserte sus datos en la tabla de hombres si es hombre, o en la tabla de mujeres si es mujer.
3. Crea un TRIGGER que se encargue de, DESPUÉS de que se produzca la actualización de un paciente, realice lo siguiente:
  - a. Si antes estaba en la tabla de HOMBRES y ahora es MUJER:
    - i. Que lo borre de la tabla de HOMBRES
    - ii. Que lo inserte en la tabla de MUJERES
  - b. Si antes estaba en la tabla de MUJERES y ahora es HOMBRE:
    - i. Que lo borre de la tabla de MUJERES
    - ii. Que lo inserte en la tabla de HOMBRES

4. En la tabla CONTADOR\_VISITAS nos encontramos con el nº de visitas que ha realizado cada paciente. Crea un TRIGGER que se encargue de, DESPUÉS de que inserte una nueva visita para un paciente, incremente su contador de visitas en 1.
5. Crea un TRIGGER que se encargue de, ANTES de que se borre una visita para un paciente (imaginemos que se creó por error), decremente su contador de visitas en 1.

### Ejercicio 7.2.

Dentro de la BBDD “mundo” vamos a implementar los siguientes PROCEDIMIENTOS y FUNCIONES (algunos/as utilizarán CURSORES):



1. Crea una tabla (pais\_estadisticas) en la que vamos a almacenar el CODE de cada país (que será su clave primaria y a la vez clave ajena a la tabla PAIS), y en la cual vamos a almacenar la siguiente información:

- ✓ Número de idiomas hablados en el país (numidiomas INT)
- ✓ Cantidad de ciudades que tiene dicho país (numciudades INT)

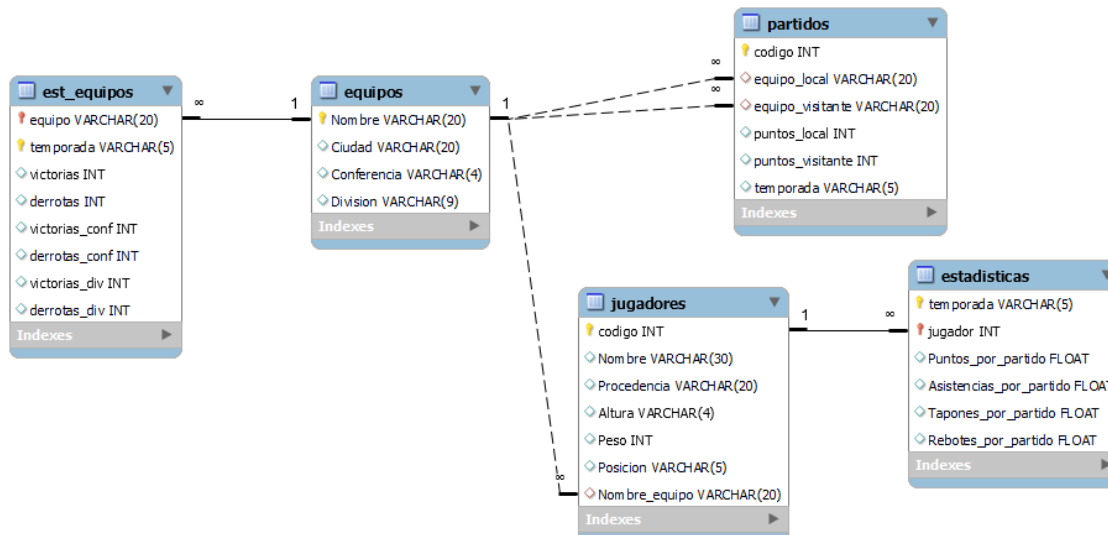
Ahora crea un PROCEDURE que recorra los datos, para cada país, de los idiomas (y cargue el nº de idiomas hablados en el campo numidiomas de la nueva tabla), y luego cuente el nº de ciudades en ese país y lo cargue en el campo numciudades de la nueva tabla.

Este PROCEDURE debe borrar, al inicio del mismo, los datos que tengamos en la nueva tabla, para asegurarnos de que no vamos a provocar un fallo de integridad referencial al cargar los datos en la tabla.

2. Crea una función que calcule la edad media de las poblaciones de un país (f\_calc\_media\_pobl). Mediante una consulta (SELECT), muestra el nombre de cada una de las ciudades de España, su población, la población media de las ciudades de España, y la resta de ambas cantidades.

### Ejercicio 7.3.

Dentro de la BBDD nba, vamos a implementar las siguientes FUNCIONES y PROCEDIMIENTOS (algunas utilizarán CURSORES):



1. Crea un nuevo campo en la tabla “jugadores” que se llame Alturacm DECIMAL(5,2), e insértalo después del campo “Altura” y antes de “Peso”.

Crea un procedimiento que, en primer lugar, borre los datos de la Alturacm de la tabla “jugadores”, y que luego calcule, para cada jugador, su altura en centímetros en función de su altura en pies y pulgadas. Dado que el campo “Altura” es de tipo CHAR, debes realizar la conversión de los valores uno a uno, aplicando el valor correspondiente en “Alturacm” en función del valor que tengas en “Altura”. Debes realizar un UPDATE distinto en función del valor. Por ejemplo:

```
UPDATE jugadores SET alturacm = 165.1 WHERE paltura = "5-5";
```

2. Aplica una mejora al procedimiento anterior. Investiga si puedes cambiar el formato del contenido del campo “altura”, para poder obtener un número decimal al cual se le pueda aplicar directamente la fórmula de conversión de pies a centímetros. Realiza un nuevo procedimiento paltura2 que haga las mismas operaciones. Haz uso de un cursor para recorrer los jugadores.
3. Crea un PROCEDURE (pcalc\_est\_equipos) que grabe datos en la tabla est\_equipos, y los datos que tiene que grabar son las victorias y derrotas totales, de la temporada 99/00, de todos los equipos de la nba. Para hacer este procedimiento hay que tener en cuenta los siguientes detalles:
  - Los datos a alimentar son los del nombre del equipo (est\_equipos.equipo), que viene de equipos.nombre, la temporada (99/00), el nº de victorias y el nº de derrotas.

- El consejo para hacer este PROCEDURE es: (1) crear un cursor que recorra todos los equipos. (2) Para cada equipo recuperado en este cursor, (2.1) crear otro cursor que recorra todos sus partidos de la temporada 99/00 como LOCAL o como VISITANTE, que calcule si el partido lo ha ganado o lo ha perdido, y le suma un 1 a una variable "vict" (si ha ganado) o a una variable "derr" (si ha perdido). OJO que los puntos local o los puntos visitante serán los que le toque según sea local o visitante, por lo tanto hay que tener esto en cuenta a la hora de ver si ha ganado o perdido, como local o como visitante
  - Una vez acabado este segundo cursor, hay que insertar en est\_equipos los datos que tocan.
4. Crea una función que calcule el peso medio de todos los jugadores según su posición. Utiliza esta función para realizar una consulta (SELECT) que recupere los nombres y el peso de todos los jugadores de la NBA, y que muestre la diferencia entre su peso y el peso medio de los jugadores de su posición. Incluye una sentencia que nos lo permita probar (una SELECT con Pau Gasol, donde veamos su peso, el peso promedio según su posición, y la diferencia entre ambas).
  5. Crea una función llamada faltmediaequip, que calcule la altura media de todos los jugadores según el equipo en el que estén (según alturacm). Utiliza esta función para realizar una consulta (SELECT) que recupere los nombres de todos los equipos de la NBA, y que utilice esta función para recuperar la altura media.
  6. Crea una función llamada fperjug, que recibe como parámetros el ID de un jugador, una temporada, que calcule el PER y devuelva este número. El PER se calcula mediante la suma de puntos por partido + rebotes por partido + asistencias por partido + robos por partido / 2 (datos en tabla "estadísticas"). De igual manera, obtén el PER de Kobe Bryant
  7. Crea una función llamada faltmepar que, recibiendo como parámetros la "procedencia" de un jugador, y su posición, devuelva la altura media de los jugadores con estas características. Usar "alturacm". Haz la prueba con los jugadores en posición Forward "F", y procedencia "Florida".
  8. (PARA MATRÍCULA). Reutiliza el PROCEDURE del punto 2 y modifica el código del mismo para que haga lo siguiente: en primer lugar, que borre los datos de la tabla est\_equipos. Después, para todos los equipos, de todas las temporadas, grabar sus victorias y derrotas TOTALES de cada temporada (campos victorias, derrotas), sus victorias y derrotas de cada temporada contra enemigos de su misma conferencia (campos victorias\_conf, derrotas\_conf), y sus victorias y derrotas de cada temporada contra enemigos de su misma división (campos victorias\_div, derrotas\_div).