EXAMEN 2ª EVALUACIÓN **BASES DE DATOS**
17/02/2022

Nombre: ----------

Apellidos: ----------

1. Una compañía aseguradora de tipo sanitario desea que se le diseñe una primera versión de una base de datos para informatizar parte de su gestión hospitalaria.

   Los hospitales de su red pueden ser propios o concertados. De todos los hospitales se debe almacenar el código del hospital (CodHos), su nombre (NomHos) y número de camas (NumCam). Ademas, cuando el hospital es propio se debe almacenar el presupuesto (Presupuesto) y tipo de servicio (TipSer), y cuando es concertado las características (Caracteristicas).

   Para los hospitales interesa saber cuáles son sus médicos, así como la fecha desde la cual trabajan con ellos. Un médico puede trabajar en varios hospitales.

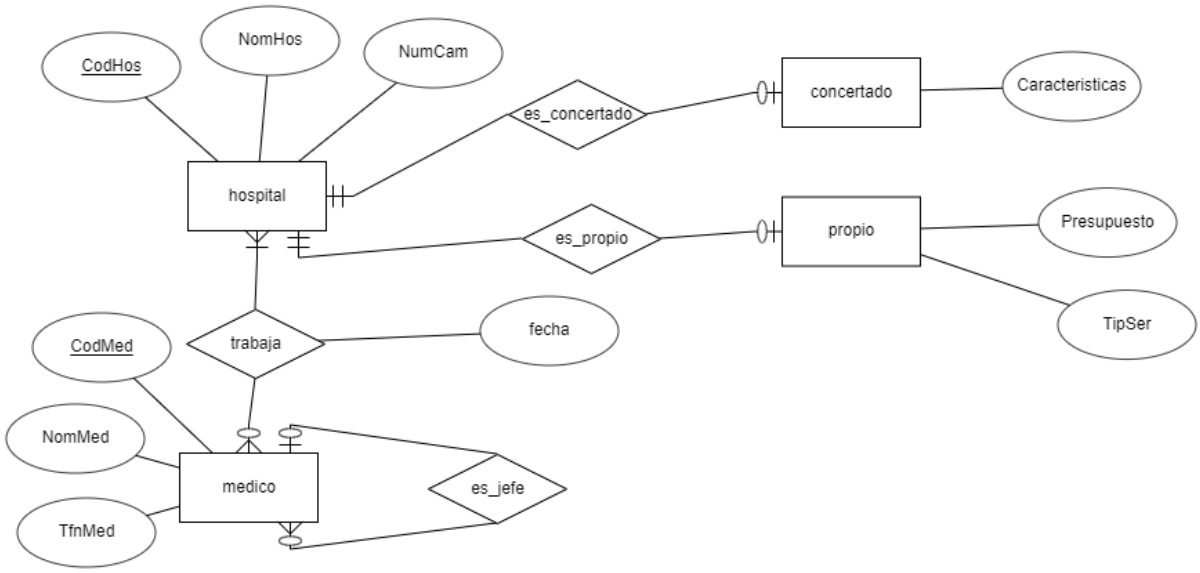
   Los médicos que se identifican por su código (CodMed), tienen un nombre (NomMed) y un teléfono de contacto (TfnMed). Existe una dependencia jerárquica entre médicos y de forma que un médico tiene un único jefe o ninguno. Un médico puede ser jefe de muchos otros o de ninguno.

   Es posible tener dado de alta un hospital y que durante algún tiempo no tenga médicos asignados al mismo.

   Un médico dado de alta trabajará al menos para un hospital.

   Se pide utilizando exactamente los nombres que se indican para los atributos:

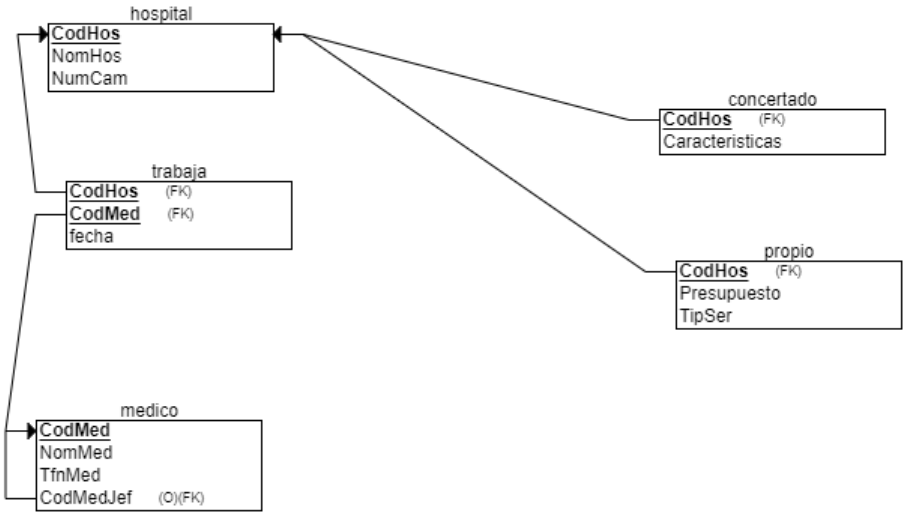
   A. Realizar el Diagrama Entidad – Relación. Debes usar el tipo de correspondencia 1 a 0.



B. Convertir el modelo anterior a un modelo relacional (DED) (Antes de entregar esta primera parte, debes quedarte con una copia de este apartado).

(Antes de entregar esta primera parte, debes quedarte con una copia de este apartado).

(2 puntos)



Este diagrama os puede servir para haceros una idea, pero se pide el DED, no el diagrama referencial.

EXAMEN 2ª EVALUACIÓN **BASES DE DATOS**
17/02/2022

Nombre: ----------

Apellidos: ----------

2. Crear una base de datos llamada como tú (si fuera mi examen se llamaría EVA). Crea todas las tablas obtenidas en el ejercicio anterior, menos las correspondientes para almacenar la información adicional si el hospital es concertado y si es hospital propio. Pero, la tabla de hospital tendrá además de los atributos que tú has modelado, cuatro atributos adicionales: TipoHos, Presupuesto (admite valores nulos), TipSer (admite valores nulos) y Caracteristicas (admite valores nulos). Las tablas deben cumplir lo siguiente:

   - En cada tabla debes definir la clave primaria. En caso de que una tabla haya surgido de una relación muchos a muchos, define la clave primaria compuesta.
   - Para las claves foráneas debes hacer siempre la definición de estas en la sentencia de creación de la tabla, junto con sus correspondientes reglas de modificación y borrado (cualesquiera que sean). La regla de modificación deben ser siempre restringida. La de borrado, deben ser como sigue:
     - Si hay un intento de borrar un médico que ha trabajado en algún hospital, no se debe permitir.
     - Si hay un intento de borrar un hospital para el cual hay almacenados médicos que trabajan en él, se deben borrar en cascada.
     - Si hay un intento de borrar un médico que es jefe de algún otro, no se debe permitir.
   - El tipo de datos para el teléfono debe ser CHAR(9) y debe cumplir que empiece por 6, 7 o 9 y tener nueve dígitos.
   - Los códigos son de tipo entero. Además, define el código de los hospitales (solo este) para que se autoincremente.
   - Utiliza la restricción adecuada para verificar que el atributo TipoHos toma el valor CONCERTADO, o bien el valor PROPIO. Además, por defecto el valor que toma es PROPIO.
   - Cualquier otra especificación no indicada debes decidirla tú mismo con coherencia según lo aprendido en clase.

   (1 punto)

```
CREATE DATABASE EVA
GO
USE EVA
GO
CREATE TABLE hospital
(
  CodHos INT IDENTITY PRIMARY KEY,
  NomHos VARCHAR(100) NOT NULL,
  NumCam SMALLINT NOT NULL,
  TipoHos VARCHAR(10) NOT NULL CHECK (TipoHos IN ('CONCERTADO', 'PROPIO')) DEFAULT 'PROPIO',
  Presupuesto MONEY NULL,
  TipSer VARCHAR(80) NULL,
  Caracteristicas VARCHAR(200) NULL
 );
 CREATE TABLE medico
(
  CodMed INT NOT NULL,
  NomMed VARCHAR(50) NOT NULL,
  TfnMed VARCHAR(9) NOT NULL CHECK (TfnMed LIKE '[679][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
  CodMedJef INT,
  PRIMARY KEY (CodMed),
  FOREIGN KEY (CodMedJef) REFERENCES medico(CodMed)
                                      ON DELETE NO ACTION
                                      ON UPDATE NO ACTION
);
CREATE TABLE trabaja
(
  Fecha DATE NOT NULL,
  CodHos INT NOT NULL,
  CodMed INT NOT NULL,
  PRIMARY KEY (CodHos, CodMed),
  FOREIGN KEY (CodHos) REFERENCES hospital(CodHos)
                                ON DELETE CASCADE
                                   ON UPDATE NO ACTION,
  FOREIGN KEY (CodMed) REFERENCES medico(CodMed)
                                ON DELETE NO ACTION
                                ON UPDATE NO ACTION
);
```

3. Utilizando sentencias del LMD de SQL, inserta filas en todas las tablas. De manera que insertes:
   - Tres hospitales. Los dos primeros hospitales son concertados. El tercer hospital es propio y debe tener un Presupuesto de 4000 euros.
   - Cinco médicos. Los dos primeros son jefes (y no tiene jefe). El primero es jefe del tercero y del cuarto. Y el segundo, del quinto.
   - En el primer hospital trabajan los médicos primero, tercero y cuarto. Y en el segundo hospital, trabajan los médicos segundo y quinto. La fecha de alta del primer médico debe ser 17/10/2010. Y la del segundo, 10/01/2021.
     (0,5 puntos)

```
INSERT INTO hospital
VALUES ('Hopital1', 500,'CONCERTADO', 5000,'Servicio1', 'Car1'),
       ('Hopital2', 1000,'CONCERTADO', 10000,'Servicio1','Car2'),
       ('Hopital3', 200,DEFAULT, 4000,'Servicio2','Car3')

INSERT INTO medico
VALUES (1,'Medico1','623456789',NULL),
       (2,'Medico2','623456780',NULL),
          (3,'Medico3','623456781',1),
          (4,'Medico4','623456782',1),
          (5,'Medico5','623456783',2)

INSERT INTO trabaja
VALUES ('17/10/2010',1,1),
       ('10/01/2021',1,3),
       ('10/02/2019',1,4),
       ('01/02/2020',2,2),
       ('01/02/2019',2,5)
```

4. Utilizando el LMD de SQL, modifica los hospitales que tienen más de dos médicos trabajando, aumentándoles el presupuesto un 2%. La sentencia debe estar dentro de una transacción y cuando compruebes que la modificación se ha realizado correctamente, debes deshacerla. ***(1 punto)***

```
BEGIN TRANSACTION
UPDATE hospital
SET Presupuesto = Presupuesto*1.02
WHERE (SELECT COUNT(*) FROM trabaja  WHERE CodHos=hospital.CodHos)>2
ROLLBACK TRANSACTION
```

5. Utilizando el LMD de SQL, borra los hospitales con presupuesto inferior a 5000 euros que no tengan aún médicos trabajando en ellos. La sentencia debe estar dentro de una transacción y cuando compruebes que la modificación se ha realizado correctamente, debes deshacerla. ***(1 punto)***

```
BEGIN TRANSACTION
DELETE FROM hospital
WHERE Presupuesto < 5000  AND CodHos NOT IN (SELECT CodHos FROM trabaja)
ROLLBACK TRANSACTION
```

6. Realiza la siguiente consulta en SQL: obtener el nombre de los hospitales concertados que tienen algún médico trabajando en los mismos que en el presente año tengan una antigüedad superior a dos años. (Esta consulta debe funcionar todos los años sin tener que cambiar nada) ***(1,5 puntos)***

```
SELECT DISTINCT NomHos
FROM hospital h JOIN trabaja t ON (h.CodHos=t.CodHos)
WHERE TipoHos LIKE 'CONCERTADO' AND (YEAR(GETDATE()) - YEAR(fecha))>2
```

7. Realiza la siguiente consulta en SQL: obtener el nombre de los médicos que son jefes, trabajan en algún hospital concertado y han sido dado de alta entre los años 2020 y 2022. Utiliza un predicado BETWEEN. ***(1,5 puntos)***

```
SELECT NomMed
FROM (medico m JOIN trabaja t ON (M.CodMed=T.CodMed)) JOIN hospital  h ON (h.CodHos=t.CodHos)
WHERE CodMedJef IS NULL AND YEAR(fecha) BETWEEN 2020 AND 2021
```

8. Utilizando el LDD de SQL, modificar la estructura de la tabla donde se almacenan los médicos que trabajan en cada hospital, de manera que la fecha tome por defecto la fecha del día actual. ***(0,25 puntos)***

```
ALTER TABLE trabaja
ADD CONSTRAINT rest1 DEFAULT GETDATE() FOR fecha
```

9. Crea un índice sobre el nombre de los hospitales. ***(0,25 puntos)***

```
CREATE INDEX INDICE1
ON hospital (NomHos)
```

10. Crea una vista que al consultarla aparezca por cada hospital el nombre de este y cuantos médicos tiene trabajando. Si un hospital aún no tiene médicos almacenados, el nombre del hospital debe salir igualmente y el número de médicos debe ser 0. Consulta la vista. Además de poder consultar la vista anterior, ¿qué otras sentencias del lenguaje de manipulación de datos de SQL se pueden realizar sobre la vista del ejercicio? ***(1 punto)***

```
CREATE VIEW vista1 (Hospital,Número_Medicos)
AS
SELECT NomHos, COUNT(t.CodHos)
FROM hospital h LEFT JOIN trabaja t ON (h.CodHos=t.CodHos)
GROUP BY NomHos


SELECT * FROM vista1
```

NOTA: Debes entregar un fichero cuyo nombre tenga la siguiente forma: primerapellido_segundoapellido_nombre.sql (por ejemplo, si fuera mi examen el fichero, se llamaría Perales_Belizón_Eva.sql). Este fichero debe contener en su interior tu nombre y apellido, además del número de cada pregunta.