

Bases de Datos
Curso 2015-2016
Grado en Ingeniería del Software
Examen Junio

Nombre: _____

Se debe entregar esta hoja

1) (4 puntos) A partir de la información sobre la BD que se describe más abajo, se pide:

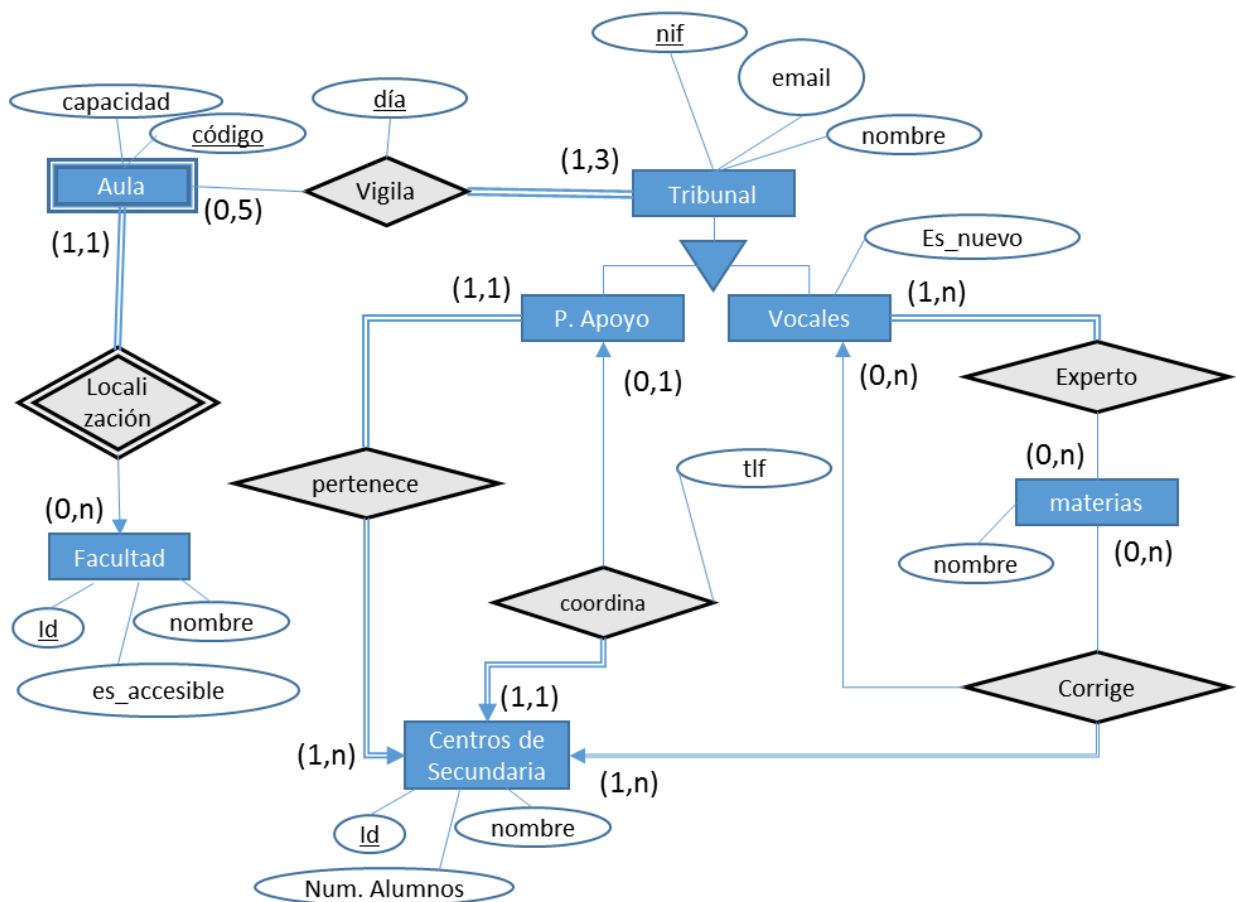
- a) (2 puntos) El esquema entidad-relación, incluyendo atributos, claves, restricciones de cardinalidad y participación.
- b) (1,5 puntos) Pasar al modelo relacional haciendo uso de las transformaciones apropiadas e indicando las restricciones de integridad referencial resultantes.
- c) (0,5 puntos) Indicar otras restricciones de integridad del modelo que no se hayan podido implementar en los modelos anteriores, o que se hayan perdido en la transformación.

Se desea diseñar la base de datos para el sistema informático de gestión y organización de las pruebas de acceso a la universidad que se celebrarán en la UCM.

- Las pruebas de acceso se celebrarán los días 7, 8 y 9 de junio y tendrán lugar en algunas de las facultades de la UCM. De las facultades se desea registrar la siguiente información: identificador de centro, nombre de la facultad y si las instalaciones tienen accesibilidad o no.
- Por otro lado, se desea registrar la información de los centros de enseñanza secundaria de la Comunidad de Madrid que realizarán las pruebas en facultades de la UCM. De los centros se desea registrar el identificador, nombre del centro y el número de alumnos que realizarán las pruebas. Cada centro designa a uno de sus profesores como coordinador y responsable del centro durante las pruebas. Se desea registrar el nombre y el teléfono de contacto de dicho coordinador para solicitar ayuda en caso de incidencias durante las pruebas.
- En cuanto a los tribunales correctores de las pruebas, se desea almacenar la información de todos los miembros que lo componen. Estos son los vocales (personal docente de la universidad) y el personal de apoyo (docentes de los centros de enseñanza secundaria). De éstos últimos, nos interesa saber si son coordinadores o no. Para todos ellos se almacenará el nombre, el NIF y el email. Para el personal de apoyo en aula se desea conocer el centro al que pertenecen. Para los vocales se desea registrar información de si han participado en las pruebas de acceso de años anteriores y la(s) materias (Matemáticas II, Geografía, Latín, etc.) de las que son expertos. No se registrarán en la base de datos aquellos vocales que no hayan demostrado que son expertos en alguna materia. No se permitirá la participación de centros de enseñanza secundaria que no propongan al menos un profesor como personal de apoyo (podría ser el propio coordinador).
- Todos los miembros del tribunal corrector, incluidos los coordinadores, tienen la obligación de actuar como vigilantes en las aulas durante los días de la prueba.

Cada aula se identifica por un código dentro de su facultad (A1, A2, ...). También interesa conocer la capacidad de las aulas y los vigilantes asignados. En cada aula puede haber un máximo de 5 vigilantes. Cada vigilante puede vigilar un máximo de 3 aulas, una por cada día (que podría ser la misma). Por supuesto, un aula que no se utilice para las pruebas, no tendrá asignados vigilantes.

- Una vez finalizada la prueba se informa a los vocales las materias que tienen que corregir (entre las que son expertos). Cada vocal puede corregir varias materias de distintos centros (Juan puede corregir Matemáticas II en dos centros distintos y Luis puede corregir las materias de Griego y Latín del mismo centro), pero cada centro solo tiene un corrector por materia. Podría darse el caso de vocales que no tengan materias asignadas para corregir.



b) Modelo relacional

Centro(id_centro, nombre, num_alumnos)

Tribunal(nif, email, nombre)

Personal_Apoyo(nif, tlf*, id_centro)

NOTA: En principio, la relación "coordina" podría incorporarse la clave de Personal_Apoyo en Centro, ya que la participación de Centro en "coordina" es total. Sin embargo, nos obliga a comprobar que el coordinador del centro sea realmente personal de dicho centro (indicado con la clave ajena id_centro de Personal_apoyo). Para evitar esto, ponemos la clave del centro en Personal_apoyo, que ya existe (por lo que nos la podemos ahorrar), y será el atributo de la

relación "tlf" el cual tomará valor para el caso de los coordinadores y NULL para los no coordinadores.

Vocales(nif, es_nuevo)

Materias(nombre materia)

Experto(nif, nombre materia)

Facultad(id fac, nombre, es_accessible)

Aulas(id fac, código aula, capacidad)

Vigila(nif, id fac, código aula, dia)

NOTA: Si incluimos "dia" como clave entonces satisfacemos la restricción de que un profesor no puede cuidar dos exámenes el mismo día, y que sí puede cuidar el mismo aula (en días diferentes).

Corrige(materia, centro, nif) ó Corrige(materia, nif, centro)

$\Pi_{id_centro}(\text{Personal_apoyo}) \subseteq \Pi_{id_centro}(\text{Centro})$

$\Pi_{nif}(\text{Vocales}) \subseteq \Pi_{nif}(\text{Tribunal})$

$\Pi_{nif}(\text{Personal_apoyo}) \subseteq \Pi_{nif}(\text{Tribunal})$

$\Pi_{nif}(\text{Experto}) \subseteq \Pi_{nif}(\text{Vocales})$

$\Pi_{nombre_materia}(\text{Experto}) \subseteq \Pi_{nombre_materia}(\text{Materias})$

$\Pi_{id_fac}(\text{Aulas}) \subseteq \Pi_{id_fac}(\text{Facultad})$

$\Pi_{id_fac, codigo_aula}(\text{Vigila}) \subseteq \Pi_{id_fac, codigo_aula}(\text{Aulas})$

$\Pi_{nif}(\text{Vigila}) \subseteq \Pi_{nif}(\text{Tribunal})$

$\Pi_{dia}(\text{Vigila}) \subseteq \{6, 7, 8\};$

c) No es posible comprobar que:

- El vocal corrija materias de las que sea experto.

Se pierden en la transformación al relacional:

- La participación máxima (5) de aula en vigila ni la participación mínima y máxima (1 y 3) de tribunal en vigila.
- Que todos los vocales sean expertos de al menos una materia
- Que todo centro aparezca en corrige
- Que solo haya un coordinador por centro

Se pueden utilizar disparadores.

2) (6 puntos) Dado el siguiente modelo relacional

- Jugador(nick, sexo, edad)
- Partida(nickRetador, nickRetado, juego, fecha, resultado*)
- Maestro(nick, juego)

Aclaración: El atributo juego toma valores de nombres de juegos de mesa de 2 jugadores como el ajedrez, las damas, el conecta-4, etc a los que los jugadores juegan partidas.

Además considera las siguientes restricciones:

- Sexo, carácter 'H' ⇔ Hombre, 'M' ⇔ Mujer
- Las partidas tienen estas restricciones:
 - Un jugador no puede jugar una partida contra sí mismo.
 - La fecha de la partida será la fecha de finalización de la misma.
 - El resultado es un entero que indica el vencedor 1=retador, 2=retado y 0=tablas. Si la partida está en curso el resultado es NULL.
- En la tabla Maestro se guardan aquellos jugadores que son ‘maestros’ en un determinado juego. Esta condición se alcanza tras ganar más de 20 partidas de ese juego.

Creación de tablas y restricciones de integridad

- a) (1 punto) Escribe las sentencias SQL que permitan crear las tablas del modelo relacional propuesto incluyendo las restricciones de integridad referencial y las otras restricciones de integridad descritas.

```
CREATE TABLE Jugador (  
  nick VARCHAR(10) PRIMARY KEY  
  , sexo CHAR(1) NOT NULL  
  , edad INTEGER NOT NULL  
  , CONSTRAINT edadCK CHECK (edad > 0)  
  , CONSTRAINT sexoCK CHECK (sexo IN ('H', 'M'))  
);  
  
CREATE TABLE Partida (  
  nickRetador VARCHAR(10)  
  , nickRetado VARCHAR(10)  
  , juego VARCHAR(10)  
  , fecha DATE  
  , resultado INTEGER DEFAULT NULL  
  , PRIMARY KEY (nickRetador, nickRetado, juego, fecha)  
  , FOREIGN KEY (nickRetador) REFERENCES Jugador (nick)  
  , FOREIGN KEY (nickRetado) REFERENCES Jugador (nick)  
  , CONSTRAINT partidaCK CHECK (nickRetador <> nickRetado)  
  , CONSTRAINT resultadoCK  
    CHECK (resultado BETWEEN 0 AND 2 OR resultado is NULL)  
);  
  
CREATE TABLE Maestro (  
  nick VARCHAR(10)  
  , juego VARCHAR(10)  
  , PRIMARY KEY (nick, juego)
```

```
, FOREIGN KEY (nick) REFERENCES Jugador(nick)
);
```

b) (0,5 puntos) ¿Qué restricciones de integridad no has podido incluir en las sentencias de creación? ¿Cómo podrías implementar dichas restricciones de integridad?

No se pueden implementar:

- El hacer que una persona figure como maestro de un juego.

Para implementarlos:

- Se pueden utilizar disparadores.

Realizar las siguientes consultas SQL

c) (0,5 puntos) Juego al que se han empatado más partidas.

```
SELECT P1.juego
FROM Partida P1
WHERE P1.resultado = 0
GROUP BY P1.juego
HAVING COUNT(*) >= ALL ( SELECT COUNT(*)
                        FROM Partida P2
                        WHERE P2.resultado = 0
                        GROUP BY P2.juego )
```

d) (0,5 puntos) Número de partidas que ha jugado cada “retador” contra adversarios más jóvenes que él.

```
SELECT P.nickRetador, count(*)
FROM
  (Jugador J1 INNER JOIN Partida P ON J1.nick = P.nickRetador)
  INNER JOIN Jugador J2 ON P.nickRetado = J2.nick
WHERE
  J1.edad > J2.edad
GROUP BY P.nickRetador
```

e) (0,5 puntos) Maestros de un juego que han jugado partidas a otros juegos.

```
SELECT DISTINCT M.nick, M.juego
FROM Maestro M
WHERE EXISTS (SELECT *
             FROM Partida P
             WHERE P.juego <> M.juego and
                  (M.nick = P.nickRetador or M.nick = P.nickRetado))

SELECT DISTINCT M.nick, M.juego
FROM Maestro M, Partida P
WHERE (M.nick=P.nickRetador OR M.nick = P.nickRetado) AND
      (M.juego<>P.juego)
```

f) (0,75 puntos) Nick, edad y número de juegos en los que son maestros aquellas personas que han jugado más de 100 partidas (a cualquier juego).

```
SELECT nick, edad, count(*)
FROM Jugador INNER JOIN Maestro USING (nick)
WHERE 100 <(SELECT count(*)
           FROM Partida P
```

```

WHERE P.nickRetador = nick OR P.nickRetado=nick)
GROUP BY nick, edad

SELECT J.nick, J.edad, count(*)
FROM Jugador J, Maestro M, Partida P
WHERE M.nick=J.nick AND J.nick=P.nickRetador OR J.nick = P.nickRetado
GROUP BY J.nick, J.edad
HAVING count(*)>100

```

g) (0,75 puntos) Nick de las personas que sólo han retado al ajedrez, es decir, que no han jugado como retadores en otro juego que no sea el ajedrez.

```

SELECT DISTINCT P1.nickRetador
FROM Partida P1
WHERE P1.juego = 'ajedrez' AND
      NOT EXISTS (SELECT *
                  FROM Partida P2
                  WHERE
                    P2.juego <> 'ajedrez'
                    AND P1.nickRetador = P2.nickRetador )

SELECT DISTINCT P1.nickRetador
FROM Partida P1
WHERE P1.juego = 'ajedrez'
      AND P1.nickRetador NOT IN (SELECT nickRetador
                                FROM Partida P
                                WHERE P.juego <> 'ajedrez')

```

h) (0,75 puntos) Número de partidas ganadas por el retador, por el retado y empatadas para cada juego.

```

SELECT * FROM
(SELECT P1.juego, count(*) AS "g-retador" FROM Partida P1 WHERE
resultado=1 GROUP BY P1.juego)
FULL OUTER JOIN
(SELECT P2.juego, count(*) AS "g-retado" FROM Partida P2 WHERE
resultado=2 GROUP BY P2.juego) USING (JUEGO)
FULL OUTER JOIN
(SELECT P3.juego, count(*) AS "tablas" FROM Partida P3 WHERE
resultado=0 GROUP BY P3.juego) USING (JUEGO)

SELECT juego, resultado, count(*)
FROM Partida
GROUP BY juego, resultado
ORDER BY 1,2;

```

i) (0,75 puntos) Haz un disparador que dada una actualización de la partida para fijar el vencedor, compruebe si el vencedor tiene que ser considerado maestro o no, según las indicaciones dadas en el enunciado.

```

CREATE OR REPLACE TRIGGER update_partida_maestro
AFTER UPDATE OF resultado ON Partida
FOR EACH ROW
DECLARE
vencedor Partida.nickRetador%TYPE;

```

```

ganadas INTEGER;
BEGIN
  IF :new.resultado = 1 THEN
    vencedor := :old.nickRetador;
  ELSE
    IF :new.resultado = 2 THEN
      vencedor := :old.nickRetado;
    END IF;
  END IF;

  IF :new.resultado <> 0 THEN
    SELECT count(*) INTO ganadas FROM partida
      WHERE juego = :old.juego and
        (nickRetador= vencedor AND resultado=1) OR
        (nickRetado= vencedor AND resultado=2);

    IF ganadas=20 THEN
      INSERT INTO Maestro (vencedor, :old.juego);
    END IF;
  END IF;
END;
/

```