

 Universidad Politécnica de Madrid ETSI SISTEMAS INFORMÁTICOS	Apellidos:		
	Nombre:		
DNI:	Num. mat.:	Grupo:	

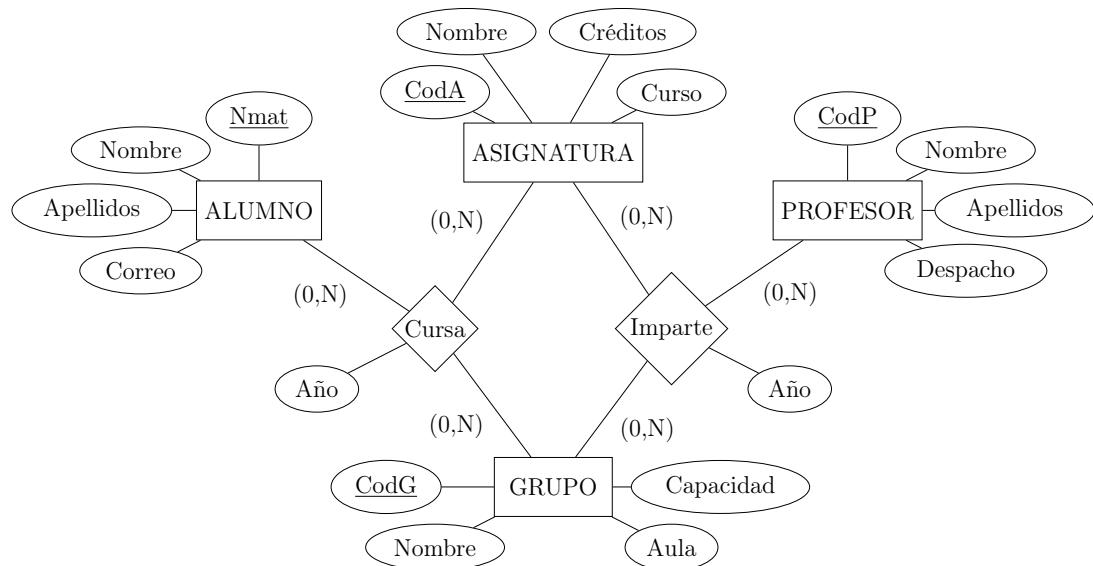
Ejercicio:	Modelado	Consultas	Gestión	Ficheros	Total:
Puntos:	2	3	3	2	10
Nota:					

Normativa de examen

- No está permitido el uso de dispositivos móviles ni otros dispositivos electrónicos, así como libros ni apuntes.
- Durante el examen, los profesores podrán solicitar acreditar la identidad de los participantes en el mismo. Deberá tener en todo momento su Documento Nacional de Identidad y/o Carné de la UPM visible sobre la mesa.
- Deberá escribir su nombre, con bolígrafo, en todas las hojas de las que consta el examen.
- No se permite abandonar el aula de examen durante los primeros 15 minutos. Transcurrido este tiempo, no se permitirá entrar al examen.
- Este examen consta de 4 ejercicios para un total de 10 puntos.
- El examen tiene una duración máxima de **2.5 horas**.
- Justifique sus respuestas lo mejor posible indicando, si fuese necesario, los pasos realizados.
- Las calificaciones provisionales serán publicadas en el Moodle de la asignatura el día 25 de enero de 2021.
- La fecha para la revisión del examen se anunciará en el Moodle de la asignatura.

1. (2 Puntos) Modelado

La *ETSISI* cuenta con un sistema para gestionar la información de qué alumnos se encuentran matriculados en cuales asignaturas y a qué grupo están asignados en cada caso. También se lleva la gestión de qué profesores imparten cuales asignaturas en cuales grupos en particular. Esto es útil, entre otras cosas, para gestionar las encuestas de alumnos a profesores. El modelo de datos que soporta este sistema en la actualidad es el siguiente:



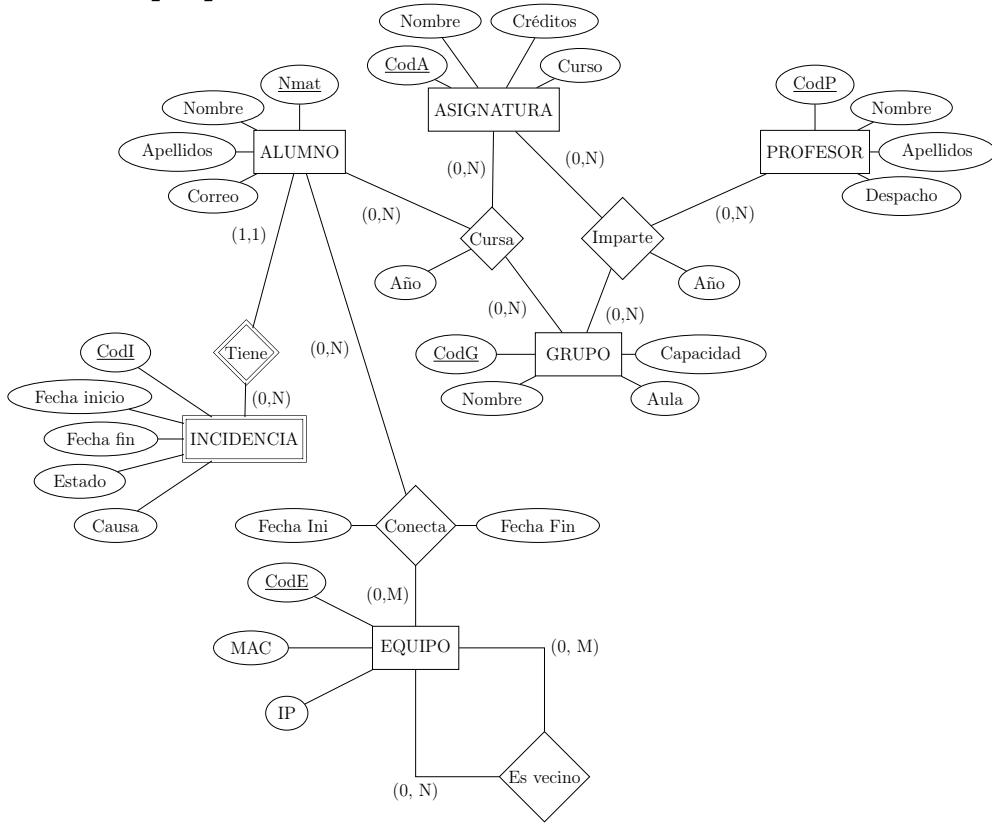
A consecuencia de la *COVID*, es necesario mantener una trazabilidad de qué alumnos están sentados junto a cuáles durante las clases presenciales. De esta forma, si un alumno informa de que ha contraído la enfermedad, se debe informar a los alumnos que han estado sentados a su lado durante alguna clase en la última semana. También los profesores que le hayan impartido docencia deben ser informados. Para ello, los alumnos tienen que iniciar sesión en el ordenador que tienen en su puesto en las clases presenciales. Existe un sistema que captura la fecha y hora de inicio y fin de sesión, a qué usuario corresponde (login y password) y desde qué máquina (MAC y dirección IP). Se tiene información por cada máquina de qué otras máquinas tiene a su alrededor. Al cabo de dos semanas, los alumnos que eran positivos pasan a considerarse en situación normal y se debe notificar al profesor que el alumno ya puede acudir a las clases presenciales.

Además, hay alumnos que informan de que están en cuarentena, pero no son positivos confirmados. En estos casos, no se informará a los alumnos que se sentaron cerca, pero sí al profesor, indicado la causa, para que tome las medidas académicas necesarias (realización de exámenes, retransmisión de clases presenciales, etc.). Es estos casos, también se informará al profesor al cabo de dos semanas (o si el alumno presenta PCR negativo) de que el alumno es considerado en situación normal.

Se pide: Hacer las modificaciones necesarias al modelo de datos inicial para poder llevar a cabo estas labores de control y trazabilidad de alumnos positivos *COVID*. Se pueden incluir nuevas entidades y relaciones.

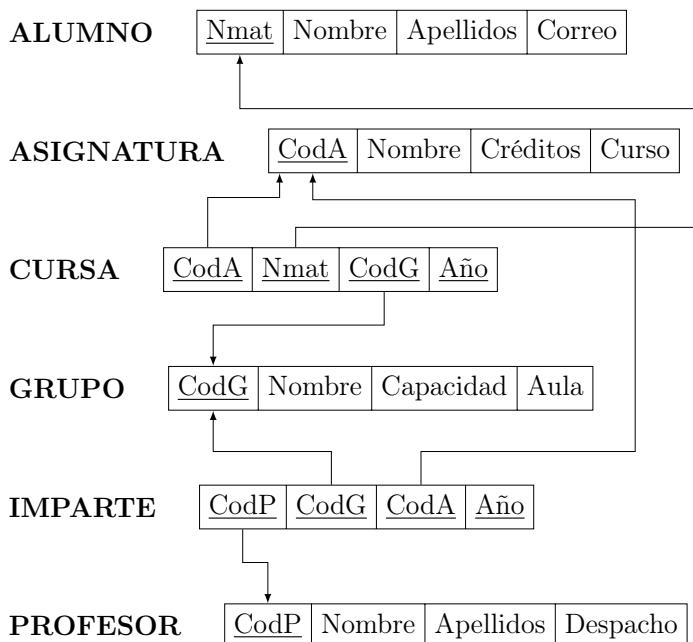
IMPORTANTE: Se penalizará la inclusión de elementos no necesarios o redundantes.

Solución propuesta:



2. (3 puntos) Consultas

Utilizando el modelo relacional que resulta de aplicar la transformación a tablas al modelo entidad-relación **sin modificar** del ejercicio 1, resuelva las siguientes consultas:



(a) Álgebra relacional:

- i. ($\frac{1}{2}$ Punto) Obtener los nombres y apellidos de todas aquellas personas, es decir, alumnos y/o profesores, que hayan compartido grupo y asignatura en el curso 2020.

Solución propuesta:

$$\begin{aligned} & \Pi_{Nombre, Apellidos} \left[PROFESOR \bowtie \sigma_{Año=2020} (IMPARTE) \right] \\ & \quad \cup \\ & \Pi_{Nombre, Apellidos} \left[ALUMNO \bowtie \sigma_{Año=2020} (CURSA) \right] \end{aligned}$$

- II. (½ Punto) Obtener el nombre, apellidos y código de aquellos profesores que, habiendo impartido docencia en segundo curso, no han impartido docencia en un aula con una capacidad superior a 40 alumnos.

Solución propuesta:

$$\begin{aligned} \Pi_{Nombre, Apellidos, CodP} & \left[PROFESOR \bowtie IMPARTE \right. \\ & \quad \left. \bowtie \sigma_{Curso=2} (ASIGNATURA) \right] \\ - \\ \Pi_{Nombre, Apellidos, CodP} & \left[PROFESOR \bowtie IMPARTE \right. \\ & \quad \left. \bowtie \sigma_{Capacidad > 40} (GRUPO) \right] \end{aligned}$$

(b) **SQL:**

- I. (½ Punto) Obtener el número de matrícula y correo electrónico de aquellos alumnos que hayan cursado todas las asignaturas de primer curso.

Solución propuesta:

```

SELECT Nmat, Correo
FROM ALUMNO Alu
WHERE NOT EXISTS (
    SELECT *
    FROM ASIGNATURA Asig
    WHERE Curso = 1 AND NOT EXISTS (
        SELECT *
        FROM CURSA C
        WHERE Alu.Nmat = C.Nmat AND Asig.CodA = C.CodA))

```

- II. (½ Punto) Obtener el número de matrícula, nombre, apellidos y total de créditos cursados para aquellos alumnos que hayan cursado más de 10 asignaturas.

Solución propuesta:

```
SELECT Nmat, Nombre, Apellidos, SUM(Creditos)
FROM ALUMNO INNER JOIN CURSA ON ALUMNO.Nmat = CURSA.Nmat
    INNER JOIN ASIGNATURA ON CURSA.CodA = ASIGNATURA.CodA
GROUP BY Nmat, Nombre, Apellidos
HAVING COUNT(CodA) > 10
```

- III. (½ Punto) Obtener los nombres y apellidos de todas aquellas personas, es decir, alumnos y/o profesores, que hayan compartido grupo y asignatura en el curso 2020.

Solución propuesta:

```
SELECT Nombre, Apellidos
FROM ALUMNO INNER JOIN CURSA ON ALUMNO.Nmat = CURSA.Nmat
WHERE Año = 2020

UNION

SELECT Nombre, Apellidos
FROM PROFESOR INNER JOIN IMPARTE ON PROFESOR.CodP = IMPARTE.CodP
WHERE Año = 2020
```

IV. (½ Punto) **Consulta Inversa:** ¿Qué enunciado tendría como solución la siguiente consulta?

```
SELECT Nmat, SUM(Creditos)
FROM ASIGNATURA a INNER JOIN Cursa c ON a.CodA = c.CodA
GROUP BY c.Nmat
HAVING SUM(Creditos) >= ALL (
    SELECT SUM(Creditos)
    FROM ASIGNATURA a INNER JOIN Cursa c ON a.CodA = c.CodA
    GROUP BY c.Nmat
)
```

Solución propuesta:

Obtener el número de matrícula y el total de créditos para aquellos alumnos que hayan cursado el mayor número de créditos de toda la base de datos.

3. (3 puntos) **Gestión**

- (a) (1 Punto) Diseñar una función que, pasándole como parámetro el número de matrícula de un alumno, devuelva un valor booleano True si dicho alumno ha cursado todas las asignaturas que están registradas en la base de datos, y False en caso contrario.

Solución propuesta:

```
DELIMITER ##
CREATE FUNCTION examen (alum INTEGER)
RETURNS BOOLEAN
DETERMINISTIC
BEGIN
    DECLARE res INTEGER;
    SELECT COUNT(*) INTO res
    FROM (SELECT Nmat, SUM(Creditos)
          FROM ASIGNATURA a INNER JOIN Cursa c ON a.CodA = c.CodA
          GROUP BY c.Nmat
          HAVING SUM(Creditos) >= ALL (
              SELECT SUM(Creditos)
              FROM ASIGNATURA))
    WHERE Nmat = alum;
    IF res = 0 THEN
        RETURN(FALSE);
    ELSE
        RETURN(TRUE);
    END IF
END##
DELIMITER ;
```

- (b) (1 Punto) Diseñar un trigger que implemente la siguiente funcionalidad: cuando un profesor deje de impartir docencia en un grupo, habrá que desmatricular automáticamente a todos los alumnos de dicho grupo, **uno a uno** (por lo tanto es **obligatorio** usar cursos).

Solución propuesta:

```
DELIMITER //
CREATE TRIGGER examen AFTER DELETE ON IMPARTE
FOR EACH ROW
BEGIN
    DECLARE alu INTEGER;
    DECLARE c CURSOR FOR
        SELECT Nmat
        FROM CURSA
        WHERE CodG IN (
            SELECT CodG
            FROM IMPARTE
            WHERE CodP = OLD.CodP);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
    OPEN c;
    read_loop: LOOP
        FETCH c INTO alu;
        DELETE FROM CURSA WHERE Nmat = alu;
        IF done THEN
            LEAVE read_loop;
        END IF;
    END LOOP;
    CLOSE c;
END//
DELIMITER ;
```

(c) (1/2 Punto) **Índices**

- I. ¿Qué indica el número 10 en la siguiente consulta de creación de un índice en SQL?

```
CREATE INDEX mi_index ON mi_tabla (nombre(10));
```

Solución propuesta:

Ese número indica que el índice que se está creando se define sobre los 10 primeros caracteres de los valores de la columna nombre.

- II. ¿Qué índices se crean automáticamente en MySQL/MariaDB?

Solución propuesta:

Cada vez que se crea una tabla y se define una o varias columnas como clave primaria, se genera un índice sobre dichas columnas de manera automática. También se crean índices automáticamente sobre las columnas que se definan como clave ajena.

- III. ¿Cuáles son las desventajas de definir índices sobre columnas de una tabla?

Solución propuesta:

Cuando creamos un índice sobre las columnas de una tabla y, posteriormente, ejecutamos instrucciones de modificación (INSERT, DELETE, UPDATE), todos los índices asociados a dicha tabla tienen que actualizarse. Por lo tanto, se ralentizan las operaciones de escritura sobre las tablas que tengan índices.

- (d) (1/2 Punto) Explica qué es una transacción en el contexto de las bases de datos y da un ejemplo de su uso. Además, nombra y explica las propiedades que deben cumplir todas las transacciones de una base de datos.

Solución propuesta:

Es una agrupación de operaciones sobre una base de datos que se ejecutan de forma atómica, aislada, mantienen la consistencia de la base de datos y los cambios realizados son duraderos. Un posible ejemplo sería una base de datos para reflejar las transacciones bancarias.

Propiedades ACID:

- Atomicidad: instrucciones se ejecutan de manera autónoma.
- Consistencia: los cambios deben dejar la base de datos en un estado consistente.
- Isolation (aislamiento): las instrucciones se ejecutan de manera aislada, sin interdependencias.
- Durabilidad: los cambios realizados deben ser durables en el tiempo y tolerantes a fallos.

4. (2 puntos) Ficheros

- (a) (1 Punto) Dado el siguiente fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<weatherdata>
    <location>
        <name>Madrid</name>
        <country>ES</country>
    </location>
    <forecast>
        <time day="2020-12-16">
            <temperature min="5.41" max="11.21"></temperature>
            <pressure unit="hPa" value="1015"></pressure>
            <humidity value="70" unit "%"></humidity>
        </time>
        <time day="2020-12-17">
            <temperature min="7.78" max="11.5"></temperature>
            <pressure unit="hPa" value="1023"></pressure>
            <humidity value="67" unit "%"></humidity>
        </time>
        <time day="2020-12-18">
            <temperature min="5.5" max="11.15"></temperature>
            <pressure unit="hPa" value="1021"></pressure>
            <humidity value="70" unit "%"></humidity>
        </time>
        <time day="2020-12-19">
            <temperature min="6.91" max="8.18"></temperature>
            <pressure unit="hPa" value="1017"></pressure>
            <humidity value="92" unit "%"></humidity>
        </time>
    </forecast>
</weatherdata>
```

- I. ¿Cuál es el resultado exacto de la siguiente consulta XQuery?

```
for $x in /weatherdata/forecast/time
where $x/temperature/@min > 6.0
return <fecha>{$x/@day}</fecha>
```

Solución propuesta:

```
<fecha day="2020-12-17"/>
<fecha day="2020-12-19"/>
```

- II. Diseña una consulta en XQuery para obtener información sobre la fecha del día en el que se ha alcanzado la mayor presión atmosférica (*pressure*). El resultado de la consulta debe ser el siguiente: 2020-12-17

Solución propuesta:

```
for $x in /weatherdata/forecast/time
where $x/pressure/@value = max(/weatherdata/forecast/time/pressure/@value)
return $x/@day
```

(b) (1 Punto) Utilizando el modelo relacional que resultaría de aplicar la transformación a tablas al modelo entidad-relación **sin modificar** del ejercicio 1 (ver ejercicio 2), construya un fichero JSON en el cual se reflejen las siguientes instancias de entidades y relaciones. El fichero deberá usar un diseño normalizado (referenciado).

- Alumno: Nmat=aa123, Nombre=Donald, Apellidos=Trump, Correo=d@dt.com
- Asignatura: CodA=1111, Nombre='Bases de Datos', Créditos=9, Curso=2
- Grupo: CodG=IIM1, Nombre=Mañana, Aula=3203, Capacidad=40
- Profesor: CodP=4321, Nombre=Edgar, Apellidos=Codd, Despacho=9001
- Cursa: Nmat=aa123, CodA=1111, CodG=IIM1, Año=2020
- Imparte: CodP=4321, CodA=1111, CodG=IIM1, Año=2020

Solución propuesta:

```
{
  "Alumnos" : [
    {
      "Nmat": "aa123",
      "Nombre": "Donald",
      "Apellidos": "Trump",
      "Correo": "d@dt.com"
    }
  ],
  "Asignaturas": [
    {
      "CodA": 1111,
      "Nombre": "Bases de Datos",
      "Créditos": 9,
      "Curso": 2
    }
  ],
  "Grupos": [
    {
      "CodG": "IIM1",
      "Nombre": "Mañana",
      "Aula": 3202,
      "Capacidad": 40
    }
  ],
  "Profesores": [
    {
      "CodP": 4321,
      "Nombre": "Edgar",
      "Apellidos": "Codd",
      "Despacho": 9001
    }
  ],
  "Cursa": [{"Nmat": "aa123", "CodA": 1111, "CodG": "IIM1", "Año": 2020}],
  "Imparte": [{"CodP": 4321, "CodA": 1111, "CodG": "IIM1", "Año": 2020}]
}
```