

PARCIAL 1: UD2 Y UD3

Sea el siguiente esquema relacional que almacena información relativa a las distintas maratones que se corren en todo el mundo. El esquema relacional es el siguiente:

CIUDAD(cod_ciu: char(15), nombre: char(50), país: char(20), historia: char(200))

CP: {cod_ciu}

VNN: {nombre, país}

MARATÓN(cod_mar: entero, nombre: char(20), fecha: date, cod_ciu: char(15), edición: entero)

CP: {cod_mar}

VNN: {fecha, edición, cod_ciu}

CAj: {cod_ciu} → CIUDAD

Único: {cod_ciu, edición}

CORREDOR(dni: char(10), nombre: char(40), edad: entero, país: char(20), sexo: char(1))

CP: {dni}

VNN: {nombre, edad, país, sexo}

HA_CORRIDO(cod_mar: entero, dni: char(10), tiempo: time)

CP: {cod_mar, dni}

CAj: {cod_mar} → MARATÓN

CAj: {dni} → CORREDOR

VNN: {tiempo}

INCIDENCIA(cod_mar: entero, num: entero, descripción: char(100), km: entero)

CP: {cod_mar, num}

CAj: {cod_mar} → MARATÓN

VNN: {km, descripción}

Donde las relaciones tienen el siguiente significado:

- **Ciudad:**
 - *cod_ciu*: código de la ciudad
 - *nombre*: nombre de la ciudad
 - *país*: país donde está la ciudad
 - *historia*: breve reseña histórica
- **Maratón:**
 - *cod_mar*: código de la maratón
 - *nombre*: nombre de la maratón
 - *fecha*: cuándo se corre la maratón
 - *cod_ciu*: código de la ciudad donde se corre la maratón
 - *edición*: vez que se ha corrido la maratón en la ciudad
- **Corredor:**
 - *dni*: DNI del corredor
 - *nombre*: cómo se llama
 - *sexo*: {H,M} Hombre o mujer
 - *edad*: cuántos años tiene
 - *país*: país de origen
- **Ha corrido:** el corredor de DNI *dni* ha corrido la maratón de código *cod_mar* en *tiempo*.
- **Incidencia:**
 - *cod_mar*: código de maratón
 - *num*: número de la incidencia
 - *descripción*: qué ha pasado
 - *km*: en qué kilómetro ha pasado

Resuelva las siguientes cuestiones:

- 1) En que consisten la independencia lógica y la independencia física en un sistema de bases de datos. (0,5 puntos)
- 2) Escriba en SQL las instrucciones que permitirían resolver las siguientes consultas:
 - a) Obtener el dni y el nombre del corredor más joven de Italia. (0'5 puntos)
 - b) Obtener el código y el nombre de las maratones que no han tenido incidencias y en las que han corrido más de 1000 mujeres . (0'75 puntos)
 - c) Obtener dni y el nombre de los corredores que han ganado al menos dos maratones. (Obviamente una maratón la gana el corredore que la ha corrido en menor tiempo) (puntos 0'75)
 - d) Obtener para cada maratón que hay en la base de datos en la que la edad media de sus corredores sea menor de 40 años, el código, el nombre, y la cantidad de incidencias que ha habido en la maratón. (0'75 puntos)
 - e) Obtener el dni y el nombre de los corredores que han corrido en todas las maratones en las que ha corrido el corredor de dni '00X' (si es que ha corrido alguna). (0'75 puntos)

SOLUCIONES

1)

- Independencia lógica entre el esquema lógico y los esquemas externos: los esquemas externos y los programas de aplicación no deben verse afectados por modificaciones del esquema lógico referentes a datos que no utilizan.
- Independencia física entre esquema lógico y esquema interno: el esquema lógico no debe verse afectado por cambios en el esquema interno referentes a la implementación de las estructuras de datos, los modos de acceso, el tamaño de las páginas y otros detalles de representación física.

2)

a)

```
SELECT C.dni, C.nombre
FROM Corredor C
WHERE C.país='Italia' AND C.edad = (SELECT MIN(C1.edad)
                                   FROM Corredor C1
                                   WHERE C1.país='Italia');
```

b)

```
SELECT M.cod_mar, M.nombre
FROM Maraton M
WHERE M.cod_mar NOT IN (SELECT I.cod_mar FROM Incidencia I) AND
  (SELECT COUNT(*)
   FROM Ha_corrido H, Corredor C
   WHERE C.dni=H.dni AND H.cod_mar=M.cod_mar AND
         C.sexo='M')>1000;
```

c)

```
SELECT C.dni, C.nombre
FROM Corredor C
WHERE (SELECT COUNT(*)
      FROM Ha_Corrido H
      WHERE H.dni=C.dni AND H.tiempo = (SELECT MIN(H1.tiempo)
                                         FROM Ha_Corrido H1
                                         WHERE H.cod_mar=H1.cod_mar))>1;
```

d)

```
/*Solución 1*/
SELECT M.cod_mar, M.nombre, COUNT(I.num)
FROM Maraton M LEFT JOIN Incidencia I ON M.cod_mar=I.cod_mar
GROUP BY M.cod_mar, M.nombre
HAVING M.cod_mar IN (SELECT M1.cod_mar
                   FROM Ha_corrido H, Corredor C
                   WHERE H.dni=C.dni
                   GROUP BY M1.cod_mar
                   HAVING AVG(C.edad)>40)
```

```

/*Solución 2*/
SELECT M.cod_mar, M.nombre, (SELECT COUNT(I.num) FROM
                                FROM Incidencia I
                                WHERE M.cod_mar=I.cod_mar)
FROM Maraton M
WHERE M.cod_mar IN (SELECT M1.cod_mar
                    FROM Ha_corrido H, Corredor C
                    WHERE H.dni=C.dni
                    GROUP BY M1.cod_mar
                    HAVING AVG(C.edad)>40))

```

e)

```

/*Solución 1*/
SELECT C.dni, C.nombre
FROM Corredor C
WHERE NOT EXISTS (SELECT *
                  FROM Ha_corrido H
                  WHERE H.dni='00X' AND
                        NOT EXISTS (SELECT *
                                   FROM Ha_corrido H1
                                   WHERE H1.dni=C.dni AND
                                         H1.cod_mar=H.cod_mar))
AND EXISTS (SELECT *
            FROM Ha_corrido H2
            WHERE H2.dni='00X')
AND C.dni<>'00X';

```

```

/*Solución 2*/
SELECT C.dni, C.nombre
FROM Corredor C
WHERE (SELECT COUNT(*)
      FROM Ha_corrido H
      WHERE H.dni='00X')
      =
      (SELECT COUNT(*)
      FROM Ha_corrido H1
      WHERE H1.dni=C.dni AND
            H1.cod_mar IN (SELECT H2.cod_mar
                          FROM Ha_corrido H2
                          WHERE H2.dni='00X'))
AND EXISTS (SELECT *
            FROM Ha_corrido H2
            WHERE H2.dni='00X')
AND C.dni<>'00X';

```