

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

CP: {dni}

VNN: {nombre, edad, país, sexo}

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

CP: {cod_mar, dni}

CAj: {cod_mar} → MARATÓN

CAj: {dni} → CORREDOR

VNN: {tiempo}

INCIDENCIA(num: entero, cod_mar: 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}
 - *edad*: cuántos años tiene
 - *país*: país de origen
- **Incidencia:**
 - *cod_mar*: código de maratón
 - *num*: número de la incidencia
 - *descripción*: qué ha pasado
 - *km*: en qué kilómetro ha pasado
- **Ha corrido:** el corredor de DNI *dni* ha corrido la maratón de código *cod_mar* en *tiempo*.

Resuelva las siguientes cuestiones:

- 1) Defina, concisamente, los siguientes conceptos: esquema conceptual, esquema lógico, esquema físico. (0'6 puntos)
Responda caso por caso.
- 2) Escriba en SQL las instrucciones que permitirían resolver las siguientes consultas:
 - a) Obtener, de las maratones que hayan tenido al menos dos incidencias antes del kilómetro 25, el código y el nombre de la maratón, indicando también el nombre y país de la ciudad donde se corre. (0'6 puntos)
 - b) Obtener, de las maratones en las que haya corrido al menos una mujer, el código y el nombre de la maratón, indicando también el dni y nombre de la corredora (mujer) que ha corrido la maratón en menos tiempo. (0'6 puntos)
 - c) Obtener el código y el nombre de la maratón en la que han corrido más corredores (sin importar el sexo). (0'6 puntos)
 - d) Obtener el dni y el nombre de los corredores que han corrido en un tiempo menor de 02:50:00 todas las maratones de su país de origen (si es que ha habido alguna). (0'8 puntos)
 - e) Para todas y cada una de las ciudades que hay en la base de datos de un país con al menos 10 corredores, obtener el código y el nombre, indicando cuántas maratones se corrieron en esa ciudad antes de 1 de enero de 2000. (0'8 puntos)

SOLUCIÓN

- 1)
- Esquema conceptual: descripción del sistema de información desde un punto de vista organizativo independiente del SGBD que se utilice e incluso de que se utilicen o no técnicas de bases de datos.
 - Esquema lógico: definición de la base de datos expresada en términos del modelo de datos en que se base el SGBD que se vaya a utilizar sin entrar en detalles de su representación física.
 - Esquema interno (físico): definición de la representación de la base de datos en la memoria secundaria del computador.

2.a)

```
SELECT M.cod_mar, M.nombre, C.nombre, C.país
FROM Maraton M, Ciudad, C
WHERE M.cod_ciu = C.cod_ciu AND
      (SELECT COUNT(*)
       FROM Incidencia I WHERE I.cod_mar = M.cod_mar AND km < 25) > 1;
```

-- alternativa

```
SELECT M.cod_mar, M.nombre, C.nombre, C.país
FROM Maraton M, Ciudad, C, Incidencia I
WHERE M.cod_ciu = C.cod_ciu AND I.cod_mar = M.cod_mar AND km < 25
GROUP BY M.cod_mar, M.nombre, C.nombre, C.país
HAVING COUNT(*) > 1;
```

2.b)

```
SELECT M.cod_mar, M.nombre, C.dni, C.nombre
FROM Maraton M, Ha_corrido H, Corredor C
WHERE M.cod_mar = H.cod_mar AND H.dni = C.dni AND C.sexo = 'M' AND
      H.tiempo = (SELECT MIN(H.tiempo)
                  FROM Ha_corrido H, Corredor C
                  WHERE M.cod_mar = H.cod_mar AND H.dni = C.dni AND C.sexo = 'M');
```

2.c)

```
SELECT M.cod_mar, M.nombre
FROM Maraton M, Ha_Corrido H
WHERE M.cod_mar = H.cod_mar
GROUP BY M.cod_mar, M.nombre
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM Ha_Corrido H GROUP BY H.cod_mar);
```

2.d)

```
SELECT C.dni, C.nombre
FROM Corredor C
WHERE NOT EXISTS (SELECT * FROM Maraton M, Ciudad CI
                  WHERE M.cod_ciu = CI.cod_ciu AND CI.país = C.país AND
                        NOT EXISTS (SELECT *
                                   FROM Ha_Corrido H
                                   WHERE H.dni = C.dni AND M.cod_mar = H.cod_mar AND
                                         H.tiempo < '02:50:00'))
AND EXISTS (SELECT * FROM Maraton M, Ciudad CI
            WHERE M.cod_ciu = CI.cod_ciu AND CI.país = C.país);
```

-- alternativa

```
SELECT C.dni, C.nombre
FROM Corredor C
WHERE (SELECT COUNT(*) FROM Maraton M, Ciudad CI
      WHERE M.cod_ciu = CI.cod_ciu AND CI.país = C.país)
      =
      (SELECT COUNT(*) FROM Ciudad CI, Maraton M, Ha_Corrido H
      WHERE CI.país = C.país AND CI.cod_ciu = M.cod_ciu AND H.dni = C.dni AND
            H.cod_mar = M.cod_mar AND H.tiempo < '02:50:00') AND
      (SELECT COUNT(*) FROM Maraton M, Ciudad CI
      WHERE M.cod_ciu = CI.cod_ciu AND CI.país = C.país) > 0
```

-- alternativa

```
SELECT C.dni, C.nombre
FROM Corredor C, Ciudad CI, Maraton M, Ha_Corrido H
WHERE C.país = CI.país AND CI.cod_ciu = M.cod_ciu AND M.cod_mar = H.cod_mar AND
      H.dni = C.dni AND H.tiempo < '02:50:00'
GROUP BY C.dni, C.nombre
HAVING COUNT(*) = (SELECT COUNT(*) FROM Maraton M, Ciudad CI
                   WHERE M.cod_ciu = CI.cod_ciu AND CI.país = C.país);
```

2.e)

```
SELECT C.cod_ciu, C.nombre, COUNT(M.cod_mar)
FROM Ciudad C LEFT JOIN Maraton M ON C.cod_ciu=M.cod_ciu AND M.fecha<'01-01-2000'
WHERE C.país IN (SELECT X.país FROM Corredor X GROUP BY X.país HAVING COUNT(*) >=10)
GROUP BY C.cod_ciu, C.nombre;
```

-- alternativa

```
SELECT C.cod_ciu, C.nombre, COUNT(M.cod_mar)
FROM Ciudad C LEFT JOIN Maraton M ON C.cod_ciu=M.cod_ciu AND M.fecha<'01-01-2000'
GROUP BY C.cod_ciu, C.nombre, C.país
HAVING C.país IN (SELECT X.país FROM Corredor X GROUP BY X.país HAVING COUNT(*) >=10)
```

-- alternativa

```
SELECT C.cod_ciu, C.nombre, COUNT(M.cod_mar)
FROM Ciudad C, Maraton M
WHERE C.cod_ciu=M.cod_ciu AND M.fecha<'01-01-2000'
GROUP BY C.cod_ciu, C.nombre, C.país
HAVING (SELECT COUNT(*) FROM Corredor X WHERE C.país = X.país)>=10
UNION
SELECT C.cod_ciu, C.nombre, 0
FROM Ciudad C
WHERE (SELECT COUNT(*) FROM Corredor X WHERE C.país = X.país)>=10 AND
      C.cod_ciu NOT IN (SELECT M.cod_ciu FROM Maraton M WHERE M.fecha<'01-01-2000');
```

-- alternativa que cae fuera de lo que os hemos explicado y que ilustra otro tipo de soluciones.

```
SELECT C.cod_ciu, C.nombre, (SELECT COUNT(M.cod_mar) FROM Maraton M
                             WHERE C.cod_ciu=M.cod_ciu AND M.fecha<'01-01-2000')
FROM Ciudad C
WHERE C.país IN (SELECT X.país
                FROM Corredor X
                GROUP BY X.país
                HAVING COUNT(*)>=10)
```