

PARCIAL 1: UD2 Y UD3

Sea el siguiente esquema relacional que almacena información relativa a los diferentes tour que organiza una agencia de viajes. El esquema relacional es el siguiente:

PAIS (cod_pais: char(10), nombre: char(15), info: char(150))

CP:{cod_pais}

VNN:{nombre}

CIUDAD (cod_ciu: char(10), nombre:char(15), info: char(150), cod_pais: char(10))

CP:{cod_ciu}

CA:{cod_pais}→PAIS

VNN:{nombre}

TOUR (cod_to: char(5), descripcion: char(50), fecha_salida: date, fecha_llegada: date, precio: number, capacidad: number)

CP:{cod_to}

VNN:{descripcion, fecha_salida, precio}

HOTEL (cod_hotel: char(10), nombre: char(20), categoria: number, cod_ciu: char(10))

CP:{cod_hotel}

CA:{cod_ciu}→CIUDAD

VNN:{nombre, categoria, cod_ciu}

RI: categoria in (1, 2, 3, 4, 5)

VISITA(cod_to: char(5), cod_ciu: char(10), fecha_llegada: date, fecha_salida: date, cod_hotel: char(10))

CP:{cod_to,cod_ciu}

CA:{cod_ciu}→CIUDAD

CA:{cod_to}→TOUR

CA:{cod_hotel}→HOTEL

VNN:{fecha_llegada, fecha_salida}

CLIENTE (DNI: char(10), nombre: char(50), edad: number, direccion: char(30), cod_ciu: char(10))

CP:{DNI}

CA:{cod_ciu}→CIUDAD

VNN:{nombre, edad, direccion, cod_ciu}

VIAJA (cod_to: char(5), DNI_cliente: char(10))

CP: {cod_to, DNI_cliente}

CA:{cod_to}→TOUR

CA:{DNI_cliente}→CLIENTE(dni)

Donde las relaciones tienen el siguiente significado:

- **País:**
 - *cod_pais*: código del país
 - *nombre*: nombre del país
 - *info*: breve información del país
- **Ciudad:**
 - *cod_ciu*: código de la ciudad
 - *nombre*: nombre de la ciudad
 - *info*: breve información de la ciudad
 - *cod_pais*: código del país donde se encuentra la ciudad
- **Tour:**
 - *cod_to*: código del tour
 - *descripción*: descripción del tour
 - *fecha_salida*: fecha de salida del tour
 - *fecha_llegada*: fecha de regreso del tour
 - *precio*: precio del tour
 - *capacidad*: máximo número de clientes que se pueden apuntar
- **Hotel**
 - *cod_hotel*: código del hotel
 - *nombre*: nombre del hotel
 - *categoria*: categoría del hotel
 - *cod_ciu*: código de la ciudad donde está el hotel
- **Visita:** el tour de código ***cod_to*** visita la ciudad de código ***cod_ciudad*** llegando en ***fecha_llegada*** y saliendo en ***fecha_salida*** siendo la estancia en la ciudad en el hotel de código ***cod_hotel***.
- **Cliente:**
 - *DNI*: DNI del cliente
 - *nombre*: nombre del cliente
 - *edad*: edad del cliente
 - *direccion*: dirección del cliente
 - *cod_ciu*: código de la ciudad del cliente
- **Viaja:** el cliente de DNI ***DNI_cliente*** viaja en el tour de código ***cod_to***

Resuelva las siguientes cuestiones:

- 1) ¿Qué es el fichero diario (journal) y qué información contiene? ¿Cómo se realiza la recuperación de la base de datos ante fallos de memoria secundaria?. (0,5 puntos)

El fichero diario es un fichero utilizado para recuperar la base de datos ante errores en memoria principal o memoria secundaria. En el diario se guarda información sobre todas las operaciones de actualización de las transacciones.

Para recuperar la base de datos ante fallos de memoria secundaria se carga la base de datos a partir de la copia de seguridad y a continuación se rehacen todas las transacciones que aparecen confirmadas en el diario desde la fecha de la copia. No es necesario deshacer las transacciones que aparecen anuladas en el diario.

- 2) Escriba en SQL las instrucciones que permitirían resolver las siguientes consultas:

- a) Obtener la descripción de aquellos tour con un precio inferior a 500€, que contengan la palabra 'lujo' en la descripción y que no tengan fecha de llegada. (0,5 puntos)

```
SELECT descripcion
FROM Tour
WHERE precio < 500 AND descripcion LIKE '%lujo%' AND fecha_llegada IS NULL
```

- b) Obtener el dni, el nombre y la edad del cliente más joven de la ciudad de nombre 'València' (puntos 0,5)

```
SELECT dni, nombre, edad
FROM Cliente C, Ciudad Ci
WHERE C.cod_ciu = Ci.cod_ciu AND Ci.nombre = 'València'
      AND edad = (SELECT MIN(edad)
                  FROM Cliente C, Ciudad Ci
                  WHERE C.cod_ciu = Ci.cod_ciu AND Ci.nombre = 'València')
```

- c) Obtener de los tours en los que se visita más ciudades, la descripción, el precio y la cantidad de ciudades que se visita. (0,75 puntos)

```
SELECT T.descripcion, precio, COUNT(*)
FROM Tour T, Visita V
WHERE T.cod_to = V.cod_to
GROUP BY T.cod_to, T.descripcion, T.precio
HAVING COUNT(*) = (SELECT MAX(COUNT(*)) FROM Visita GROUP BY cod_to)
```

- d) Obtener para todas las ciudades de la base de datos, su nombre, la cantidad de hoteles que hay en la ciudad y la categoría media de dichos hoteles. (0,75 puntos)

```
SELECT C.nombre, COUNT(H.cod_hotel), AVG(H.categoria)
FROM Ciudad C LEFT JOIN Hotel L ON C.cod_ciu = H.cod_ciu
GROUP BY C.cod_ciu, C.nombre
```

OR

```
(SELECT C.nombre, COUNT(*), AVG(H.categoria)
FROM Ciudad C, Hotel L
WHERE C.cod_ciu = H.cod_ciu
GROUP BY C.cod_ciu, C.nombre)
UNION
(SELECT nombre, 0, NULL
FROM Ciudad
WHERE cod_ciu NOT IN (SELECT cod_ciu FROM Hotel))
```

- e) Obtener el dni y el nombre de los clientes tales que en todos los tours que han realizado se ha visitado la ciudad de código 'AJR1'. (1 punto)

```
SELECT dni, nombre
FROM Cliente C
WHERE NOT EXISTS (SELECT *
                  FROM Viaja V
                  WHERE V.dni_cliente = C.dni
                  NOT EXISTS (SELECT *
                              FROM Visita Vi
                              WHERE V.cod_to=Vi.cod_to AND Vi.cod_ciu = 'AJR1'))
AND EXISTS (SELECT *
            FROM Viaja V
            WHERE V.dni_cliente = C.dni)
```

OR

```
SELECT dni, cliente
FROM Cliente C
WHERE      (SELECT COUNT(*)
            FROM Viaja V
            WHERE V.dni_cliente = C.dni)
=
            (SELECT COUNT(*)
            FROM Viaja V, Visita Vi
            WHERE V.dni_cliente = C.dni AND V.cod_to = Vi.cod_to
                AND Vi.cod_ciu = 'AJR1')
AND (SELECT COUNT(*)
     FROM Viaja V
     WHERE V.dni_cliente = C.dni) > 0
```