Laboratorio de Bases de Datos

Juan Camilo Alzate Restrepo

Práctica 2 (Procedimientos y Funciones) **Fecha de Entrega: Diciembre 15**

1. Se tienen ciudades y diferentes empresas de transporte que las conectan así:

```
CREATE TABLE conexion(
cod_ciudad1 NUMBER(8),
cod_ciudad2 NUMBER(8),
empresa_transporte VARCHAR2(10),
valor_pasaje NUMBER(8) NOT NULL CHECK(valor_pasaje > 0),
PRIMARY KEY(cod_ciudad1, cod_ciudad2, empresa_transporte)
);

INSERT INTO conexion VALUES(1, 2, 'Azul', 10);
```

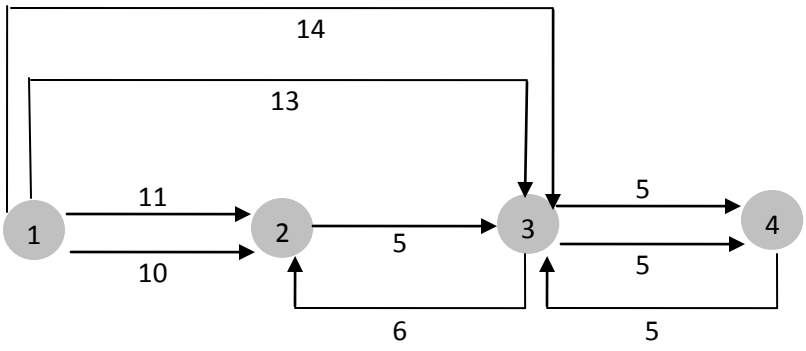
Esto significa que el pasaje entre la ciudad 1 y la ciudad 2 vale $10 en la empresa Azul.

Veamos más datos:

```
INSERT INTO conexion VALUES(1, 2, 'Gacela', 11);
INSERT INTO conexion VALUES(1, 3, 'Ondas', 14);
INSERT INTO conexion VALUES(2, 3, 'Azul', 5);
INSERT INTO conexion VALUES(1, 3, 'Azul', 13);
INSERT INTO conexion VALUES(3, 2, 'Gacela', 6);
INSERT INTO conexion VALUES(3, 4, 'Ondas', 5);
INSERT INTO conexion VALUES(3, 4, 'Azul', 5);
INSERT INTO conexion VALUES(4, 3, 'Azul', 5);
```

Etc.

Gráficamente:



Note, por ejemplo, que para un usuario ir de la ciudad 1 a la ciudad 3 dispone de dos rutas:
Ruta directa:
1→3 costo = 13 ó 14 (dependiendo de la empresa que elija).
Ruta haciendo escalas:
1→2→3 costo = 15 ó 16 (dependiendo de la empresa que elija en el tramo 1→2).

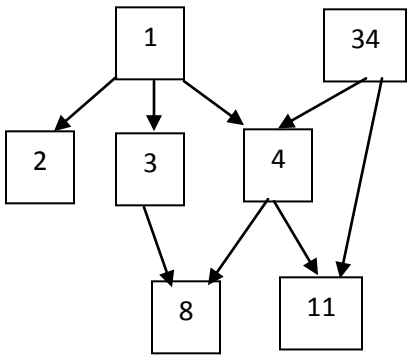
a) **(40%)** Dada una ciudad (origen) y una ciudad (destino), haga un procedimiento en PL/SQL que imprima todas las rutas posibles para ir de una ciudad a otra y el costo de cada ruta (en cada ruta no se puede repetir escala). Se debe indicar la empresa de transporte de cada tramo de la ruta. La impresión debe estar ordenada desde la ruta más económica hasta la más costosa.

Ejemplo. Si se ingresan las ciudades 1 (origen) y 3 (destino) en el ejemplo anterior, el programa debe imprimir:
1(Azul)→3, costo = 13
1(Ondas)→3, costo = 14
1(Azul)→2(Azul)→3, costo = 15
1(Gacela)→2(Azul)→3, costo = 16
Si no es posible viajar entre un par de ciudades, el programa deberá informarlo. Por ejemplo, si se ingresan las ciudades 3 (origen) y 1 (destino) en el ejemplo anterior.

Nota: Los datos anteriores son sólo un ejemplo. **Su procedimiento debe funcionar para cualquier cantidad de filas que tenga la tabla**: Puede haber muchas ciudades, muchas empresas de transporte, muchas conexiones.

2. Suponga que un producto se puede componer de muchos productos. Por ejemplo, el producto 1 se podría componer de los productos 2, 3 y 4. A su vez el producto 4 podría componerse de los productos 8 y 11, y el producto 3 podría componerse del producto 8. El producto 34 podría componerse del 4 y del 11.

Gráficamente



Sea la tabla:

```
CREATE TABLE componente(
producto NUMBER(8),
producto_que_lo_compone NUMBER(8),
PRIMARY KEY(producto, producto_que_lo_compone),
CHECK (producto <> producto_que_lo_compone)
);

INSERT INTO componente VALUES(1, 2);
INSERT INTO componente VALUES(1, 3);
INSERT INTO componente VALUES(1, 4);
INSERT INTO componente VALUES(4, 8);
INSERT INTO componente VALUES(4, 11);
INSERT INTO componente VALUES(3, 8);
INSERT INTO componente VALUES(34, 4);
INSERT INTO componente VALUES(34, 11);
```

a) **(40%)** Haga un procedimiento en PL/SQL que reciba el código de un producto e imprima todos los productos que lo componen en todos los niveles.

- *Ejemplo dado el producto 1, el programa debe imprimir:*
  Hijos primer nivel: 2, 3, 4 (son los hijos directos)
  Hijos segundo nivel: 8, 11 (son los hijos de los hijos del primer nivel)
  (Como el 8 es hijo del 3 y del 4, se imprime una sola vez en el segundo nivel).

- *Ejemplo dado el producto 34, el programa debe imprimir:*
  Hijos primer nivel: 4, 11
  Hijos segundo nivel: 8, 11
  (Note que el 11 es hijo directo e indirecto del 34, por eso sale tanto en el primer nivel como en el segundo nivel).

Nota: Los datos anteriores son sólo un ejemplo. **Su programa debe funcionar para cualquier cantidad de filas que tenga la tabla**: Puede haber muchos productos, muchos componentes.

3. Realice una función que reciba los siguientes parámetros:
FUNCTION estacontenido(xa,ya,xb,yb,xc,yc,xd,yd,x,y) RETURN boolean IS ...

- Cuatro puntos que delimitan una región **A**(xa,ya), **B**(xb,yb), **C**(xc,yc) y **D**(xd,yd)
- Un punto **P**(x,y)

a) **(20%)** Su función deberá determinar si el punto P está o no contenido dentro de la región dada.