**PRIMER TRABAJO DE BASES DE DATOS 2 (25%)**

Por favor lea todo el enunciado y los ejemplos.

Анастасия Буханцова[[1]](#footnote-0) (que no sé como se pronuncia) decide montar una granja porcina llamada “*Mi Cerdito Feliz*”.

**Punto 1 (50%)** Se tiene una tabla **CERDO** así:

CREATE TABLE cerdo(

cod NUMBER(8) PRIMARY KEY,

nombre VARCHAR(20) NOT NULL,

pesokilos NUMBER(8) NOT NULL CHECK (pesokilos > 0)

);

Анастасия también ha adquirido unos camiones para el transporte de cerdos. La tabla **CAMION** está así:

CREATE TABLE camion(

idcamion NUMBER(8) PRIMARY KEY,

maximacapacidadkilos NUMBER(8) NOT NULL CHECK (maximacapacidadkilos > 0)

);

Resulta que a Анастасия le hacen pedidos. Los pedidos consisten simplemente en un **número de kilos** que desea algún cliente, por ejemplo, un pedido por 16 kilos de cerdos.

Анастасия requiere un informe que indique cuales cerdos van en cada camión para cumplir con el pedido. Para ello, **se debe seguir el siguiente procedimiento**: Se debe tomar el camión con mayor capacidad y se llena con cerdos lo más que se pueda (sin superar su máxima capacidad en kilos). A continuación se toma el siguiente camión de mayor capacidad y se llena igualmente lo más que se pueda (sin superar su máxima capacidad en kilos) y así se continua (tenga en cuenta que al seguir este procedimiento no siempre se podrá cumplir con la cantidad **exacta** solicitada por el cliente como lo muestra el siguiente ejemplo, ya que los cerdos **se deben enviar vivos**, es decir, **no se pueden partir.** El procedimiento indicado tampoco garantiza que se obtenga la “mejor” solución).

Veamos un ejemplo concreto. Supóngase que se tienen los siguientes cerdos:

**cod nombre pesokilos**

2 Ana Criado 3

4 Dua Lipa 3

8 Saffron 3

11 Ava Max 3

15 Esthero 8

Y suponga que se tienen los siguientes camiones:

**idcamion maximacapacidadkilos**

13 10

38 7

22 8

Suponga que un cliente hace un pedido por 16 kilos. Siguiendo el procedimiento anterior se toma primero el camión 13 (el de mayor capacidad) al cual le caben 10 kilos. Con los cerdos dados **no hay forma** de llenar este camión **exactamente** con 10 kilos, lo máximo con lo que se puede llenar **en este caso** es con 9 kilos (de hecho hay varias formas para llenarlo con 9 kilos, **elija cualquiera de ellas, pero solo una**).

Supongamos que este camión lo llenamos con estos cerdos: 2 (Ana Criado), 4 (Dua Lipa) y 8 (Saffron), o sea: 3 + 3 + 3 = **9** kilos.

A continuación, se toma el siguiente camión de mayor capacidad, es decir, el camión 22 al cual le caben 8 kilos. Como faltan 7 kilos para cumplir con el pedido, al mirar los cerdos que nos quedan (el 11 y el 15) entonces vemos que el cerdo 15 pesa 8 kilos (y aunque cabe en el camión) no sirve **porque sobrepasa el número de kilos total pedidos por el cliente**. Así, que en en este caso solo montamos en este camión al cerdo 11 (Ava Max) que pesa 3 kilos.

Elabore un programa en PL/SQL que recibe el número de kilos solicitado por el cliente. El programa **debe imprimir la solución (TODO lo que está en verde) en el siguiente formato**:

Informe para Mi Cerdito.

-----

Camión: 13

Lista cerdos: 2 (Ana Criado) 3kg, 4 (Dua Lipa) 3kg, 8 (Saffron) 3kg

Total peso cerdos: 9kg. Capacidad no usada del camión: 1kg

Camión: 22

Lista cerdos: 11 (Ava Max) 3kg

Total peso cerdos: 3kg. Capacidad no usada del camión: 5kg

-----

Total Peso solicitado: 16kg. Peso real enviado: 12kg. Peso no satisfecho: 4kg.

**La salida deber ser idéntica al formato mostrado en verde.**

**Nota 1**: Si al momento de tomar los camiones hay varios empatados en la máxima capacidad, tome cualquiera de ellos. Por ejemplo, si en el caso anterior los tres camiones tuviesen la misma capacidad, entonces tome cualquiera de ellos (llénelo lo más que se pueda), luego tome el siguiente camión (llénelo lo más que se pueda) y así sucesivamente.

**Nota 3:** Si el pedido no se puede satisfacer, por ejemplo, suponga que todos los cerdos pesan más de 5 kilos y el cliente pide solamente 3 kilos, imprima: “El pedido no se puede satisfacer”.

**Punto 2 (50%)** Suponga que se tiene la siguiente tabla:

CREATE TABLE individuo(

codigo NUMBER(8) PRIMARY KEY,

nombre VARCHAR2(20) NOT NULL,

valor NUMBER(8) NOT NULL CHECK (valor > 0),

padre NUMBER(8) REFERENCES individuo, --código del padre del inviduo

nro\_hijos NUMBER(8) NOT NULL CHECK (nro\_hijos >=0),

CHECK(padre <> codigo)

);

Elabore los *triggers* necesarios para hacer cumplir las siguientes reglas:

**a)** Cuando un individuo ingresa, **el atributo nro\_hijos debe ser 0**.

**b)** Cuando un individuo ingresa con un **padre no nulo**, entonces a su padre se le debe incrementar el atributo nro\_hijos en una unidad.

**c)** Cuando un individuo se borra y tiene padre no nulo, entonces a su padre se le debe decrementar el atributo nro\_hijos en una unidad.

**d)** Cuando un individuo se borra y tiene hijos, entonces a todos sus hijos se les debe poner su atributo padre en nulo.

**e)** Si se actualiza el atributo valor de un individuo se procede así:

* Si se trata de un incremento (incremento del atributo valor):

- El incremento debe ser mayor o igual a 5, de lo contrario se rechaza el incremento.

- Al individuo **realmente solo se le incrementa su atributo valor en 2** unidades, el resto de las unidades del incremento se le suma a cualquiera de sus hijos **directos** (**pero solo a uno de ellos**). Si el individuo no tiene hijos, entonces el resto del incremento se pierde.

* Si se trata de un decremento (decremento del atributo valor):

Al individuo simplemente se le decrementa el valor en lo indicado por el *update*.

**f)** Si se actualiza el código de un individuo (por supuesto con un código que no exista en la tabla, esto **ya** lo controla la clave primaria), entonces a **todos** sus hijos se les debe propagar el cambio para que queden con el nuevo código de su padre. Si no tiene hijos, solo se le cambia el código al individuo.

Veamos un ejemplo concreto, en rojo se van indicando los cambios que le deben ocurrir a la tabla.

La tabla inicia vacía.

Ingresa el siguiente individuo:

**codigo nombre valor padre nro\_hijos**

19 Hope Sandoval 10 NULL 0

Ingresa otro individuo así:

19 Hope Sandoval 10 NULL 0

32 Kirsty Hawkshaw 8 NULL 0

Ingresa otro individuo así:

19 Hope Sandoval 10 NULL 1

32 Kirsty Hawkshaw 8 NULL 0

64 Annabella Lwin 10 **19** 0

Ingresa otro individuo así:

19 Hope Sandoval 10 NULL 2

32 Kirsty Hawkshaw 8 NULL 0

64 Annabella Lwin 10 19 0

122 Amanda Marshall 20 **19** 0

Ingresa otro individuo así:

19 Hope Sandoval 10 NULL 2

32 Kirsty Hawkshaw 8 NULL 0

64 Annabella Lwin 10 19 1

122 Amanda Marshall 20 19 0

11 Mavvie Marcos 2 **64** 0

Se emite un *update* para ponerle 18 en el atributo valor al individuo 19, queda así:

19 Hope Sandoval 12 NULL 2

32 Kirsty Hawkshaw 8 NULL 0

64 Annabella Lwin 16 19 1

122 Amanda Marshall 20 19 0

11 Mavvie Marcos 2 64 0

Al individuo 19 se le cambia su código por el 333:

333 Hope Sandoval 12 NULL 2

32 Kirsty Hawkshaw 8 NULL 0

64 Annabella Lwin 16 333 1

122 Amanda Marshall 20 333 0

11 Mavvie Marcos 2 64 0

Se borra el individuo 64:

333 Hope Sandoval 12 NULL 1

32 Kirsty Hawkshaw 8 NULL 0

122 Amanda Marshall 20 333 0

11 Mavvie Marcos 2 NULL 0

Se emite un *update* para ponerle 4 en el atributo valor al individuo 333, queda así:

333 Hope Sandoval 4 NULL 1

32 Kirsty Hawkshaw 8 NULL 0

122 Amanda Marshall 20 333 0

11 Mavvie Marcos 2 NULL 0

**Nota 1**: Por simplicidad no se harán ni *inserts*, ni *updates*, ni *deletes* masivos. Es decir, estas operaciones solo afectarán una fila.

**Nota 2:** Para las pruebas no se harán actualizaciones a otros atributos, solo lo solicitado.

**Notas adicionales:**

* Sus soluciones deben funcionar para cualquier cantidad de filas que tengan las tablas. Los datos presentados son solo para ejemplificar.
* No modifique la estructura de las tablas dadas: no cambie los nombres ni les adicione ni les quite columnas**.**
* Puede usar: todas las estructuras de datos y **todas las tablas auxiliares que desee**.
* Para entregar por email a *[fjmoreno@unal.edu.co](mailto:fjmoreno@unal.edu.co)*, el miércoles 7 de abril 2021 hasta las 11 am.
* **No se reciben trabajos en hora posterior**. No se reciben versiones “mejoradas”.
* No se revisan trabajos enviados “por accidente” a otras cuentas de correo. Solo se califican los trabajos enviados a *[fjmoreno@unal.edu.co](mailto:fjmoreno@unal.edu.co)*.
* Junto con el código se debe incluir un informe de máximo 4 páginas (letra Arial 11) donde se explique **brevemente** cómo se solucionó cada punto. Este informe hace parte de la calificación del trabajo. **No enviar los datos de prueba que usted usó para probar su código. Solo envíe el código y el informe.**
* Grupos de **tres** personas.
* Los trabajos deben ser independientes entre los grupos. Trabajos copiados **parcial** o totalmente **así sea en un SOLO punto** se califican con 0 (cero) en su totalidad para todos los integrantes. Las soluciones presentadas deben ser originales, concebidas por cada grupo, no copiadas de medios como páginas web, foros, librerías de programas, etc. El trabajo debe ser desarrollado por los integrantes del grupo **no** por personas ajenas a él.
* El monitor les puede ayudar con aspectos técnicos pero su función **no** es hacerles la práctica **ni está autorizado** para **cambiar las condiciones del trabajo**.
* Si hay errores en el enunciado por favor informarme para corregirlo.

**Francisco Moreno**

**Marzo 17 de 2021**

1. Ver https://www.youtube.com/watch?v=UnmU2D1fX9s [↑](#footnote-ref-0)