SQL

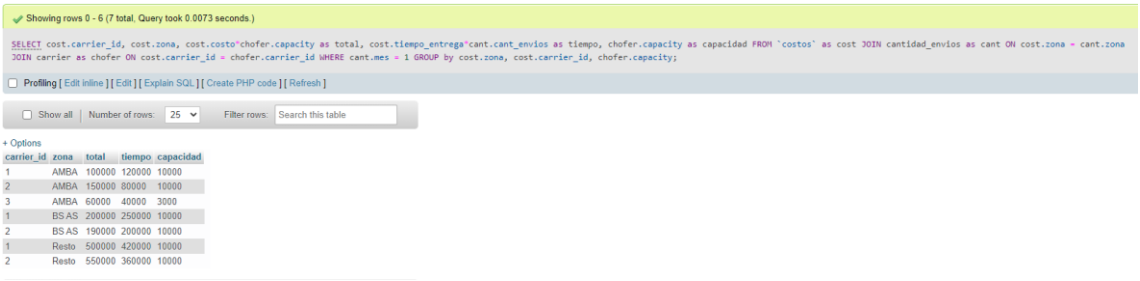
Ej 1)


```
SELECT ca.carrier_id, c.zona, ce.cant_envios/ca.capacity as veces, ca.capacity*c.costo as costo FROM `carrier` as ca JOIN costos as c ON c.carrier_id = ca.carrier_id JOIN cantidad_envios as ce ON ce.zona = c.zona GROUP BY ca.carrier_id, c.zona;
```

Dada la capacidad de cada Carrier, los envíos se deben transportar en varios meses, en caso de que la capacidad sea menor a la cantidad de envíos.

Ej 2)

```
SELECT cost.carrier_id, cost.zona, cost.costo*chofer.capacity as total, cost.tiempo_entrega*cant.cant_envios as tiempo, chofer.capacity as capacidad FROM `costos` as cost

JOIN cantidad_envios as cant

ON cost.zona = cant.zona

JOIN carrier as chofer

ON cost.carrier_id = chofer.carrier_id

WHERE cant.mes = 1 GROUP by cost.zona, cost.carrier_id, chofer.capacity;
```



Si quiero hacer la mayor cantidad de envíos en un mes:

Contemplo las 3 zonas de envíos, la máxima capacidad, el precio y tiempo de envío.

TOTAL $750.000 (con un máximo de 23000 de capacidad de entrega mensual)

Lo ideal seria poder realizar envíos de forma equitativa por zona, pero se desconoce la demanda. Serían 8000 para BSAS, 8000 para AMBA y 7000 para Resto.

Todos los Carrier deben trabajar a su maxima capacidad, por lo que el carrier 3 debe realizar las entregas de AMBA siendo estas 3000. Para completar las 5000 restantes combinamos con el Carrier 2. Finalmente, utilizamos la disponibilidad del carrier 2 sumados los 3000 del carrier

1 para completar las entregas en BSAS y nos quedarían disponibles 7000 de capacidad del carrier 1 para todo el Resto.

c3 3000 + c2 5000 = 80.000 tiempo -> AMBA

c2 5000 + c1 3000 = 175.000 tiempo -> BSAS

c1 7000 > 420.000 = 294.000 tiempo -> RESTO

TIEMPO = 549.000

Script grails

```
import me.*;
def upsPullTrkService = ctx.getBean('upsPullTrkService')
def s = Shipment.get(27528954729)
def tn = s.trackingNumber
def trackingData = upsPullTrkService.getTrkEvents([tn])
trackingData.each { td ->
  println "-----------------------------------------------"
  println "${td.sucursal} - ${td.eventDate} - ${td.description}"
}
"Done"
```

Se esta definiendo el tracking de entrega para sus datos, en donde esta comenzando a buscar procesos (o métodos, no conozco del todo grails).

Por eso se guarda el ship, el numero del envío (tracking) y cada evento que este genera (ya que, dependiendo del lugar al que llegue, este genera un reporte de su situacion con la fecha y ubicación).

Esto último lo podemos ver en el for each que hace trackingData, donde nos arroja 3 datos, la sucursal, la fecha del evento y su descripción.

Bash

Script básico bash

- A tu entender, que se busca obtener como output del script?
- Podrías detallar que se hace en cada línea del script?
- Cuántas líneas se imprimen como output?

```
#!/bin/bash
users_id=(71665538 66146765 132961968 15096445 172753273 54152646)
for users_id in ${users_id[*]}
do
        curl=$(curl -s "api.mercadolibre.com/users/$users_id/shipping_preferences" | jq -c
'.services')
        echo "$users_id: $curl"
done
```

Va a iterar sobre la lista de users pidiendo con curl esa URL.

En cada iteración reemplaza el user_id en la variable dentro de la URL.

Eso debe devolver un json, le extrae con jq el campo services y eso lo guarda en la variable curl

Después imprime user id y lo que guardó en curl

El pipe | jq -c desconozco su funcionamiento, pero entiendo que condiciona a un compact output.

Corriendo el script el output fue el siguiente: 6 líneas

```
~ » ./test.sh
71665538: [311,591,801,881,1181,136171,136501,145041]
66146765: [311,591,801,881,1181,136171,145041,155662]
132961968: [311,591,801,881,1181,136171,145041]
15096445: [311,591,801,881,1181,136171,145041,153071]
172753273: [311,591,801,881,1181,136171,145041,171441]
54152646: [311,591,801,881,1181,136171,145041,149751]
```