

<b>GESTIÓN DE DATOS</b>	<b>U.T.N. F.R.B.A.</b>
<b>Parcial 2° CUATRIMESTRE 2008</b>	<b>12/11/2008</b>

<b><u>APELLIDO</u></b>	<b><u>NOMBRE</u></b>	<b><u>N° LEGAJO</u></b>	<b><u>N° GRUPO</u></b>	<b><u>NOMBRE GRUPO</u></b>

<b><u>1-SQL</u></b>	<b><u>2-T-SQL</u></b>	<b><u>3-ALGORITMO</u></b>	<b><u>4-TP</u></b>

## **Posible Solución ej 1 y 2 Parcial de Gestión de Datos (12/11/2008)**

**1-** Escriba una consulta que retorne para todos los supervisores y vendedores (en un mismo conjunto de resultados) los siguientes datos:

Tipo: deberá retornar SUP en el caso de supervisor y VEN en el caso de los vendedores

Código: código del vendedor o supervisor según el caso

Detalle: detalle del vendedor o supervisor según el caso

Monto\_facturado: Para el caso de los vendedores suma del total neto de los comprobantes que vendió..

Para el caso de los supervisores suma del total neto de los comprobantes que vendieron los vendedores que tiene a su cargo en la actualidad dicho supervisor.

El resultado debe ser ordenado por Monto\_facturado descendente.

Se sabe que podría darse el caso en el cual un Supervisor es además vendedor. De ser así los datos están cargados de la siguiente manera: el detalle del vendedor coincide exactamente con el del supervisor y el vendedor es a su vez su propio supervisor.

Si ocurre el caso antes detallado se deberá mostrar a dicha persona solo como supervisor y no como vendedor

```
select 'SUP' Tipo, supe_detalle Detalle, isnull(sum(ccbt_neto),0)
Monto_facturado
from supervisores left join vendedores on supe_codigo=vend_supervisor left
join ccbte on ccbt_vendedor=vend_codigo
group by supe_codigo, supe_detalle
union
select 'VEN' Tipo, vend_detalle Detalle, isnull(sum(ccbt_neto),0)
Monto_facturado
from vendedores left join ccbte on ccbt_vendedor=vend_codigo
where not exists(select 1 from supervisores where supe_detalle=vend_detalle
and vend_supervisor=supe_codigo )
group by vend_codigo, vend_detalle
order by Monto_facturado desc
```

**2-** Escriba una función que dados los datos identificatorios de un cabecero de comprobante evalúe si en dicho comprobante se vendió un conjunto de productos individuales que podrían haberse facturado como un producto compuesto, en ese caso deberá retornar el código del producto compuesto que se podría haber facturado y '0000000' si no se podría haber facturado ningún compuesto.

Solo se debe considerar los casos en que se podría haber facturado un producto compuesto formado directamente por los productos vendidos.

En caso que en el comprobante se pudiera haber facturado mas de 1 compuesto deberá retornar cualquiera de ellos.

```
CREATE FUNCTION ej2 (@c char(3),@t char(1),@s char(4),@cb char(8))
RETURNS char(8)
AS
BEGIN
declare @prod char(8)

select top 1 @prod=c.comp_producto
from composicion c
where exists (select 1 from rcbt where rcbt_codigo=@c and rcbt_tipo=@t
and rcbt_sucursal=@s and rcbt_cbte=@cb and rcbt_producto=c.comp_comp and
rcbt_cantidad>=c.comp_cantidad)
group by comp_producto
having (select count(*) from composicion c2 where
c2.comp_producto=c.comp_producto)=count(*)
```

<b>GESTIÓN DE DATOS</b>	<b>U.T.N. F.R.B.A.</b>
<b>Parcial 2° CUATRIMESTRE 2008</b>	<b>12/11/2008</b>

```
RETURN isnull(@prod, '00000000')  
  
END
```