

```

/*
Mostrar para cada empleado su nombre y la cantidad de ventas
(Total $) de aquellas ordenes que superan en items/lineas el
promedio de items/lineas general de ordenes
*/

select e.FirstName,e.LastName , sum(d.UnitPrice * d.Quantity) as
total$
from Employees e
inner join Orders O on o.EmployeeID = e.EmployeeID
inner join OrderDetails D on o.OrderID = d.OrderID
group by e.FirstName,e.LastName
HAVING COUNT(*) > (select AVG(infoOrdenes.totalLineas) as
PromLinea --obtento el prm de lineas
FROM (
select d.OrderID, count(*) as
totalLineas --Orden y cant de lineas
from OrderDetails d
group by d.OrderID) as
infoOrdenes
)

/*
Mostrar el total de Compras ($) realizado que solo considere
aquellos Suppliers que unicamente participaron en compras en
Canada y aquellos que unicamente participaron en compras en UK
*/

```

```

SELECT SUM(PurchaseTotal) as Total$
FROM Purchases P
WHERE P.SupplierID IN (
(
select SupplierID
from Purchases
where PurchaseCountry = 'Canada'
EXCEPT
select SupplierID
from Purchases
where PurchaseCountry <>
'Canada'
)
UNION (
select SupplierID
from Purchases
where PurchaseCountry = 'UK'
EXCEPT
select SupplierID
from Purchases
where PurchaseCountry <> 'UK'
)
)

```

```

/*
Por cada empleado, mostrar la cantidad de $ vendidos para
aquellos empleados que su nombre incluye la letra Y, y para

```

todos los empleados, la cantidad de productos las compras  
(purchases) que realizaron  
\*/

```
select e.EmployeeID,e.FirstName,e.LastName, ( SELECT
SUM(d.UnitPrice * d.Quantity)
                                from Orders O
inner join OrderDetails D on O.OrderID = d.OrderID
inner join Employees e1 on o.EmployeeID = e.EmployeeID
where e1.EmployeeID = e.EmployeeID and e.FirstName like '%y%')
as Total$,
                                (
SELECT COUNT(DISTINCT PD.ProductID) as CantProd
From Purchases P
inner join PurchaseDetail PD on PD.PurchaseID = p.PurchaseID
WHERE P.EmployeeID = e.EmployeeID
                                )
as #ProdDistintos
from Employees e

/*
Agregar la columna "Rate" de tipo VARCHAR(30) a la tabla de
Clientes.
Actualizar el campo Rate con el valor "ORO, PLATA, BRONCE"
según: ORO: >= 30 Ordenes , PLATA Entre 15 y 29 Ordenes,
BRONCE menos de 15 ordenes
*/

ALTER TABLE Customers
ADD rate varchar(30)

UPDATE Customers SET
rate = (SELECT (CASE
                                WHEN count(o.OrderID) < 15 then 'BRONCE'
                                WHEN count(o.OrderID) BETWEEN 15 and 29
then 'PLATA'
                                ELSE 'ORO' END)
                                FROM Orders O
                                WHERE O.CustomerID = C.CustomerID)
FROM Customers C

/*
Llevar a un MR en 3FN la siguiente tabla y escribir la consulta
que muestre para cada local, la cantidad de Facturas emitidas y
```

la cantidad productos (Ids) vendidos. Mostrar los locales que no tuvieron ventas

VENTAS(numFac,numLocal, nombreLocal, fechaFac,  
codCliente,nomCliente, edadCliente,codArt,nomArticulo,  
cantComprada, PrecioUnitario)

Consideraciones: La primer factura que genera cada local es la  
nro 00001  
\*/

```
--Locales(numLocal,NombreLocal)
-----
--Clientes(codCliente,nomCliente,edadCliente)
-----
--Articulos(codArt,nomArticulo,PrecioUnitario
-----
--Facturas(numFac,numLocal,fechaFac,codCliente)
-----
--FacItems(numFac,numLocal,codArt,cantComprada,PrecioUnitario)

SELECT l.NombreLocal,COUNT(*) as TotalFacturas, COUNT(DISTINCT
FI.codArt) as #ArticulosDistintos
from Locales L
LEFT join Facturas F on F.numLocal = L.numLocal
INNER JOIN FacItems FI on FI.numFac = F.numFac and FI.numLocal =
F.numLocal
GROUP By l.NombreLocal
```

-----