

```
/*
```

1. Crear las siguientes tablas:

Charges(ChargeID, ChargeName, ChargeType)

Sabiendo que el identificador es autoincremental, el nombre no puede estar vacío y el tipo debe ser de la siguiente lista:

```
'Freight'
'Custom'
'Handle'
'Others'
```

OrdersCharges(OrderID, ChargeID, ChargeDate, ChargeAmount)

Sabiendo que la fecha no necesita almacenar el componente horas, que el monto debe ser mayor que 0 y que la clave de la tabla son los campos OrderID, ChargeID, ChargeDate

```
*/
```

```
CREATE TABLE Charges (ChargeID int identity not null,
                        ChargeName varchar(30) not null,
                        ChargeType varchar(20),
                        CONSTRAINT PK_Charges PRIMARY KEY (ChargeID),
                        CONSTRAINT CK_ChargeType CHECK (ChargeType IN
('Freight', 'Custom', 'Handle', 'Others')))
```

```
GO
```

```
CREATE TABLE OrderCharges (OrderID int not null,
                             ChargeID int not null,
                             ChargeDate date not null,
                             ChargeAmount money,
                             CONSTRAINT PK_OrderCharge PRIMARY
KEY (OrderID, ChargeID, ChargeDate),
                             CONSTRAINT FK_OrderCharge FOREIGN
KEY (OrderID) REFERENCES Orders (OrderID),
                             CONSTRAINT FK_ChargeCharges FOREIGN
KEY (ChargeID) REFERENCES Charges (ChargeID),
                             CONSTRAINT CK_Amount
CHECK (ChargeAmount > 0))
```

```
GO
```

```
/*
```

2. Cargar un registro en la tabla Charges con el tipo 'Freight', luego cargar para cada Orden un registro en la tabla OrderCharges que registre el costo por el concepto Freight, se sabe que el ChargeID es el cargo recientemente creado, que la fecha es la fecha de la orden y el monto es el Freight de la tabla Orders

```
*/
```

```
INSERT INTO Charges VALUES ('Freight Cost', 'Freight')
```

```
GO
```

```
INSERT INTO OrderCharges SELECT OrderID, (SELECT ChargeID FROM Charges
WHERE ChargeType='Freight'), OrderDate, Freight
FROM Orders
```

```
GO
```

```
/*
```

3. Para cada producto, mostrar ProductID, ProductName y la cantidad de clientes que lo compraron alguna vez, si algún cliente nunca compró alguno de los productos igual deben mostrarse los datos

```
*/
```

```
SELECT p.ProductID, p.ProductName, (SELECT count (distinct (CustomerID))
```

```
FROM Orders o, OrderDetails od
WHERE o.OrderID=od.OrderID
```

and

```
od.ProductID=p.ProductID) as CntClientes
FROM Products p
```

/*

4. Mostrar primer y segundo nombre de los empleados que prepararon las ordenes de la fecha mas antigua

*/

```
SELECT e.FirstName,e.LastName
FROM Orders o, Employees e
WHERE o.EmployeeID=e.EmployeeID and
      o.OrderDate = (SELECT MIN(OrderDate)
                     FROM Orders)
```

/*

5. Mostrar codigo y nombre de las categorías que tienen menos de 10 productos de su tipo

*/

```
SELECT c.CategoryID,c.CategoryName,count(p.productid)
FROM Categories c, Products p
WHERE c.CategoryID = p.CategoryID
GROUP BY c.CategoryID,c.CategoryName
HAVING count(p.productid) < 10
```