

1.- Utiliza la tabla Production.Product para crear una consulta que muestre los nombres de los productos y los ID's de productos que hayan sido ordenados. Utiliza una subconsulta para filtrar los productos ordenados en la tabla Sales.SalesOrderDetail. Ordena el resultado por nombre de manera ascendente.

```
SELECT p.Name,  
       p.ProductID  
FROM Production.Product p  
WHERE p.ProductID IN (SELECT sod.ProductID  
                      FROM sales.SalesOrderDetail sod)  
ORDER BY Name
```

Results			Messages		
	Name	ProductID			
1	All-Purpose Bike Stand	879			
2	AWC Logo Cap	712			
3	Bike Wash - Dissolver	877			
4	Cable Lock	843			
5	Chain	952			
6	Classic Vest, L	866			
7	Classic Vest, M	865			
8	Classic Vest, S	864			
9	Fender Set - Mountain	878			
10	Front Brakes	948			
11	Front Derailleur	945			

2.Utiliza las tablas HumanResources.Employee, HumanResources.EmployeeDepartmentHistory, HumanResources.Department y Person.Person para crear una consulta que muestre un listado de todos los empleados y el nombre de su departamento actual. Los campos a mostrar serán: BusinessEntityID, FirstName + ' ' + LastName como Nombre del Empleado, y Name como Nombre del Departamento.

```
SELECT e.BusinessEntityID,
       p.FirstName + ' ' + p.LastName AS NombreDelEmpleado,
       hrd.Name AS NombreDelDepartamento
FROM HumanResources.Employee e
INNER JOIN Person.Person p
    ON e.BusinessEntityID = p.BusinessEntityID
INNER JOIN HumanResources.EmployeeDepartmentHistory edh
    ON e.BusinessEntityID = edh.BusinessEntityID
INNER JOIN HumanResources.Department hrd
    ON edh.DepartmentID = hrd.DepartmentID
ORDER BY BusinessEntityID
```

	BusinessEntityID	NombreDelEmpleado	NombreDelDepartamento
1	1	Ken Sánchez	Executive
2	2	Teri Duffy	Engineering
3	3	Roberto Tamburello	Engineering
4	4	Rob Walters	Engineering
5	4	Rob Walters	Tool Design
6	5	Gail Erickson	Engineering
7	6	Jossef Goldberg	Engineering
8	7	Dylan Miller	Research and Development
9	8	Diane Margheim	Research and Development
10	9	Gigi Matthew	Research and Development
11	10	Michael Raheem	Research and Development

3. Utiliza las tablas HumanResources.Employee, HumanResources.EmployeeDepartmentHistory y HumanResources.Department para crear una consulta que muestre el mismo resultado que en el ejercicio 2, pero utilizando un CTE. Los campos a mostrar serán: BusinessEntityID, FirstName + ' ' + LastName como Nombre del Empleado, y Name como Nombre del Departamento.

```
WITH EmployeeDep AS (
SELECT e.BusinessEntityID,
       p.FirstName + ' ' + p.LastName AS NombreDelEmpleado,
       hrd.Name AS NombreDelDepartamento
FROM HumanResources.Employee e
INNER JOIN Person.Person p
      ON e.BusinessEntityID = p.BusinessEntityID
INNER JOIN HumanResources.EmployeeDepartmentHistory edh
      ON e.BusinessEntityID = edh.BusinessEntityID
INNER JOIN HumanResources.Department hrd
      ON edh.DepartmentID = hrd.DepartmentID
)
SELECT BusinessEntityID,
       NombreDelEmpleado,
       NombreDelDepartamento
FROM EmployeeDep
ORDER BY BusinessEntityID
```

	BusinessEntityID	NombreDelEmpleado	NombreDelDepartamento
1	1	Ken Sánchez	Executive
2	2	Teri Duffy	Engineering
3	3	Roberto Tamburello	Engineering
4	4	Rob Walters	Engineering
5	4	Rob Walters	Tool Design
6	5	Gail Erickson	Engineering
7	6	Jossef Goldberg	Engineering
8	7	Dylan Miller	Research and Development

4.Crea una tabla temporal local y otra global para almacenar los productos ordenados. Utiliza la tabla Production.Product para insertar los datos en ambas tablas temporales. Los campos a incluir serán: ProductID y Name. Inserta los datos en la tabla temporal local.

```
CREATE TABLE #TempLocal(  
ProductID INT,  
Name Varchar(255)  
);
```

```
CREATE TABLE ##TempGlobal(  
ProductID INT,  
Name VARCHAR (255)  
);
```

```
-- Insertar datos en tabla temporal local  
INSERT INTO #TempLocal(ProductID, Name)  
SELECT ProductID,  
        Name  
FROM #TempLocal
```

```
-- Insertar datos en tabla temporal Global
```

```
INSERT INTO ##TempGlobal(ProductID, Name)
```

```
SELECT ProductID,  
        Name  
FROM ##TempGlobal
```

Results		Messages
	ProductID	Name
1	1	Adjustable Race
2	879	All-Purpose Bike Stand
3	712	AWC Logo Cap
4	3	BB Ball Bearing
5	2	Bearing Ball
6	877	Bike Wash - Dissolver
7	316	Blade
8	843	Cable Lock

5.Utiliza la tabla Sales.SalesOrderHeader para crear una consulta que muestre los números de orden y números de compra solo para el año 2011.

```
SELECT SalesOrderNumber,
       PurchaseOrderNumber
FROM SALES.SalesOrderHeader
WHERE OrderDate >= 2011
```

	SalesOrderNumber	PurchaseOrderNumber
1	SO43659	PO522145787
2	SO43660	PO18850127500
3	SO43661	PO18473189620
4	SO43662	PO18444174044
5	SO43663	PO18009186470
6	SO43664	PO16617121983
7	SO43665	PO16588191572
8	SO43666	PO16008173883

6.Utiliza la tabla Sales.SalesOrderHeader para crear una consulta que muestre los números de orden y números de compra sin los 2 primeros caracteres y nombra los campos resultantes como NewSalesOrderNumber y NewPurchaseOrderNumber.

```
SELECT SUBSTRING(SalesOrderNumber, 3, LEN(SalesOrderNumber)) AS NewSalesOrderNumber,
       SUBSTRING(PurchaseOrderNumber, 3, LEN(PurchaseOrderNumber)) AS
NewPurchaseOrderNumber
FROM SALES.SalesOrderHeader;
```

	NewSalesOrderNumber	NewPurchaseOrderNumber
1	43659	522145787
2	43660	18850127500
3	43661	18473189620
4	43662	18444174044
5	43663	18009186470
6	43664	16617121983
7	43665	16588191572
8	43666	16008173883