

```
/*  
Ej. 1
```

Crear la tabla PriceHistory (Historico de Precios) donde se almacenaran los cambios de precio de los productos que hacen los empleados dicha tabla debe tener la siguiente estrucutra:

```
PriceHistory(PH_Id, ProductID, PH_Date, PH_Old, PH_New, EmployeeID)
```

Donde PH_Id es el identificador de la tabla, debe ser un numero autoincremental, PH_Date es la fecha del cambio de precios, debe poder almacenar fecha y hora, PH_Old y PH_New son campos donde se almacenará el precio anterior y el actual del producto, los campos ProductID y EmployeeID son los respectivos de sus tablas.

Crear todas las restricciones que considere necesarias para mantener la integridad referencial sabiendo tambien que el campo PH_New debe ser mayor que 0

```
*/
```

```
CREATE TABLE PriceHistory(PH_Id int identity(1,1) not null,  
                           ProductID int,  
                           PH_Date datetime,  
                           PH_Old money,  
                           PH_New money,  
                           EmployeeID int,  
                           CONSTRAINT PK_PH PRIMARY KEY(PH_Id),  
                           CONSTRAINT FK_ProdPH FOREIGN  
KEY(ProductID) REFERENCES Products(ProductID),  
                           CONSTRAINT FK_EmpPH FOREIGN  
KEY(EmployeeID) REFERENCES Employees(EmployeeID),  
                           CONSTRAINT CK_Precio CHECK (PH_new >  
0))
```

```
/*  
Ej. 2
```

Crear los indices que considere necesarios en la tabla PriceHistory

```
*/
```

```
CREATE INDEX IDX_PROD_PH ON PriceHistory(ProductID)  
CREATE INDEX IDX_EMP_PH ON PriceHistory(EmployeeID)
```

```
/*  
Ej. 3
```

Insertar 4 registros a la tabla PriceHistory

```
*/
```

```
set dateformat dmy  
INSERT INTO PriceHistory VALUES(1,'15/01/1997',0,18,1),  
                                (2,'15/07/1997',16,19,1),  
                                (10,'18/07/1997',30,31,8),  
                                (7,'19/07/1997',25,30,4)
```

```
/*  
Ej. 4
```

Mostrar codigo y descripcion de los productos que fueron ordenados en la fecha mas antigua,
no mostrar resultados repetidos

```
*/
SELECT P.ProductID,P.ProductName
FROM Products P, Orders O, OrderDetails OD
WHERE P.ProductID=OD.ProductID and
      O.OrderID=OD.OrderID and
      O.OrderDate = (SELECT MIN(OrderDate)
                     FROM Orders)
```

/*
Ej. 5

Mostrar codigo y descripcion de los Proveedores (Suppliers) que vendieron menos de 3 productos diferentes que fueron ordenados en el año 1997

```
*/
SELECT S.SupplierID,S.CompanyName
FROM Suppliers S, Products P, Orders O, OrderDetails OD
WHERE S.SupplierID=P.SupplierID and
      P.ProductID = OD.ProductID and
      O.OrderID = OD.OrderID and YEAR(O.OrderDate)=1997
GROUP BY S.SupplierID,S.CompanyName
HAVING COUNT(DISTINCT(P.ProductID))<3
```

/*
Ej. 6

Mostrar codigo, primer y segundo nombre de los Empleados, y la cantidad de Ordenes que prepararon en el año 1996
si algun empleado no preparó ninguna orden, igual deben mostrarse sus datos, ordenar de mayor a menor segun cantidad de ordenes
Debe realizar el ejercicios utilizando las dos modalidades vistas en clase (referencia externa y LEFT JOIN)

```
*/

/* Solucion con referencia externa */
SELECT E.EmployeeID,E.FirstName,E.LastName, (SELECT COUNT(O.OrderId)
                                           FROM Orders O
                                           WHERE
O.EmployeeID=E.EmployeeID and
YEAR(O.OrderDate)=1996) as Cantidad
FROM Employees E
ORDER BY Cantidad DESC
```

```
/* Solucion con LEFT JOIN*/
SELECT E.EmployeeID,E.FirstName,E.LastName,COUNT(O.OrderId)
FROM Employees E LEFT JOIN Orders O ON E.EmployeeID=O.EmployeeID and
      YEAR(O.Ordertime)=1996
GROUP BY E.EmployeeID,E.FirstName,E.LastName
```