Combinación de tablas

Combinar datos de dos o más tablas por medio de JOINS.

Introducción Teórica

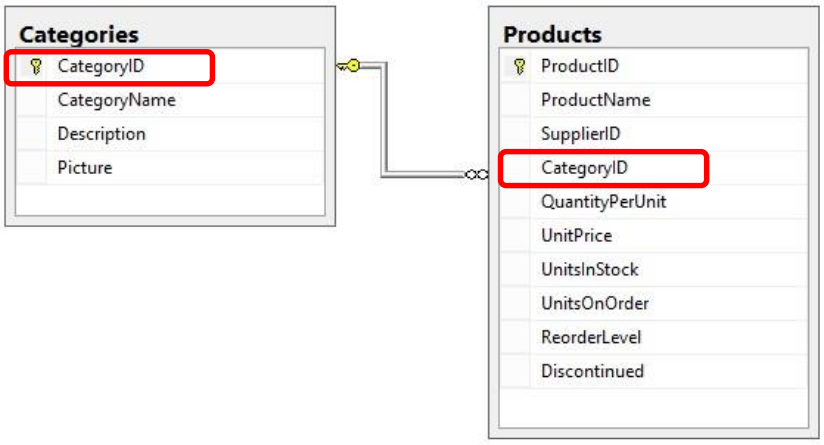
COMBINACIÓN DE TABLAS.

¿Qué se entiende por combinación de tablas?

Una combinación es una operación que permite consultar dos o más tablas para producir un conjunto de resultados que incorpore filas y columnas de cada una de las tablas en cuestión. Las tablas se combinan en función de las columnas que son comunes a ambas tablas.

El objetivo de la combinación de tablas es proporcionar al usuario datos que le permitan un fácil entendimiento de la información que requiere. Esta información, por el uso del modelo entidad – relación, se puede encontrar fragmentada en muchas tablas, y al combinarlas, se puede presentar al usuario la información pertinente de una forma más entendible.

Esta operación es conocida también como unión o vinculación de tablas.



Para los ejemplos se hace uso de la base de datos **NORTHWIND o NORTHWND**
La combinación de campos de tablas distintas sólo es posible cuando se han definido campos relacionados entre tablas. Esto es si existe un campo clave primaria en una tabla que aparece como clave foránea en la otra tabla.

La sentencia JOIN en SQL permite combinar registros de dos o más tablas en una base de datos relacional.

Sentencias JOIN

En el Lenguaje de Consultas Estructurado (SQL) hay tres tipos de JOIN: interno, externo y cruzado.

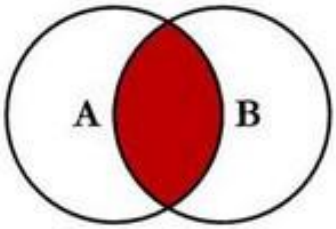
- Combinación interna INNER JOIN
- Cruzada CROSS JOIN
- Combinación externa OUTER JOIN ○ LEFT OUTER JOIN o LEFT JOIN ○ RIGHT OUTER JOIN o RIGHT JOIN ○ FULL OUTER JOIN o FULL OUTER JOIN

La palabra OUTER es opcional y no añade ninguna función

SQL JOINS

INNER JOIN

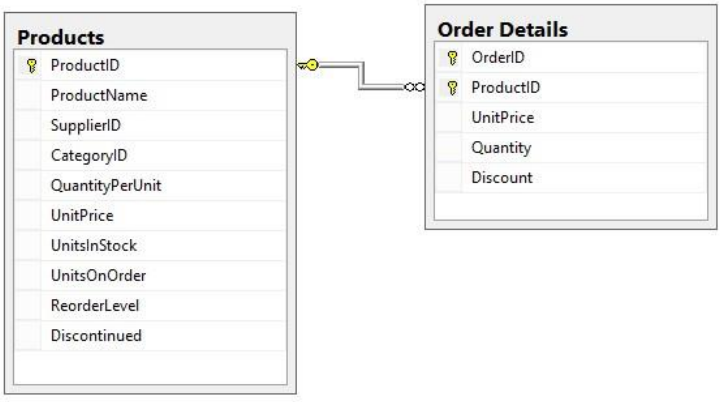
Se utiliza para mostrar los datos coincidentes entre las tablas de donde se quiere mostrar la información:




```
SELECT <lista_campos>  FROM <TablaA
A>
INNER JOIN  <TablaB B>
                    ON A.Key=B.Key
```

Nota: **ON** se utiliza para colocar los nombres de los campos con los cuales se ha realizado la relación entre las tablas.

Ejemplo 1: Se desea conocer todos los productos que se encuentran en una orden



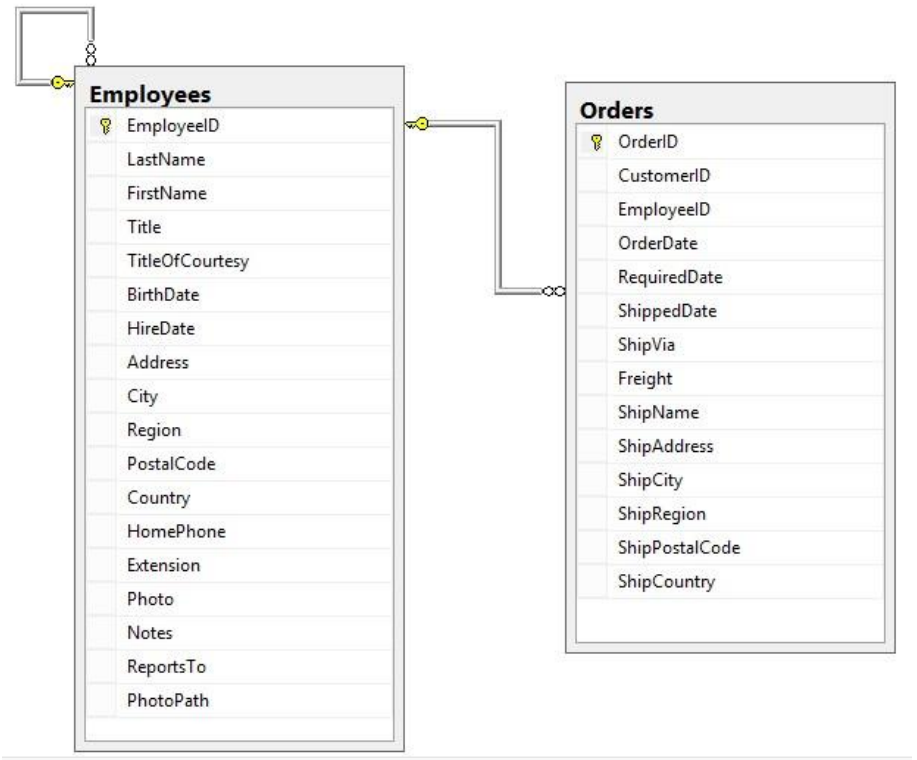
Para obtener los registros coincidentes en ambas tablas habría que realizar la siguiente consulta:

```
SELECT OrderID, P.ProductID, ProductName
FROM Products P
INNER JOIN [Order Details] OD
ON P.ProductID=OD.ProductID
```

Y se obtendría el siguiente resultado:

	OrderID	ProductID	ProductName
1	10285	1	Chai
2	10294	1	Chai
3	10317	1	Chai
4	10348	1	Chai
5	10354	1	Chai
6	10370	1	Chai
7	10406	1	Chai
8	10413	1	Chai
9	10477	1	Chai
10	10522	1	Chai

Ejemplo 2: Se desea conocer los empleados que han atendido una orden y en qué fecha lo hicieron, los registros se deben ordenar por el campo EmployeeID



La consulta SQL es:

```
SELECT LastName, Employees.EmployeeID,OrderDate FROM Orders
INNER JOIN Employees
ON Orders.EmployeeID=Employees.EmployeeID
ORDER BY Employees.EmployeeID
```

Nota: al campo EmployeeID se le coloca el nombre de la tabla de donde queremos sacar los resultados, ya que el nombre de este campo aparece tanto en la tabla Orders y Employees, y si no se utiliza así da error de nombre ambiguo.

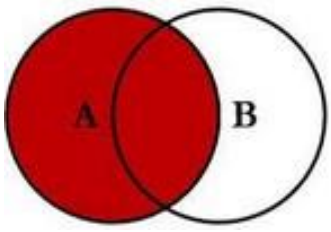
Y se obtienen los siguientes resultados:

	LastName	EmployeeID	OrderDate
1	Davolio	1	1996-07-17 00:00:00.000
2	Davolio	1	1996-08-01 00:00:00.000
3	Davolio	1	1996-08-07 00:00:00.000
4	Davolio	1	1996-08-20 00:00:00.000
5	Davolio	1	1996-08-28 00:00:00.000
6	Davolio	1	1996-08-29 00:00:00.000
7	Davolio	1	1996-09-12 00:00:00.000
8	Davolio	1	1996-09-16 00:00:00.000
9	Davolio	1	1996-09-20 00:00:00.000
10	Davolio	1	1996-09-25 00:00:00.000
11	Davolio	1	1996-09-27 00:00:00.000
12	Davolio	1	1996-10-09 00:00:00.000

Aquí se muestran las primeras 12 filas de 830 filas en total

LEFT JOIN

Muestra los registros de la tabla izquierda más los registros coincidentes con la tabla derecha




```
SELECT <lista_campos>
                FROM <TablaA A>
LEFT JOIN  <TablaB B>
                  ON A.Key=B.Key
```

Ejemplo: Se desea conocer que empleados han atendido un pedido independientemente si este lo ha realizado o no

La consulta SQL es:

```
SELECT OrderID, E.EmployeeID, Lastname
FROM Employees E
LEFT JOIN Orders O
ON E.EmployeeID=O.EmployeeID
```

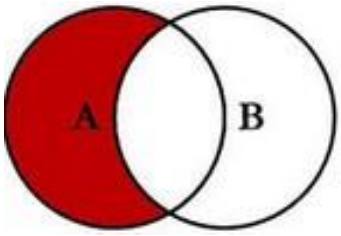
Se obtiene el siguiente resultado:

	OrderID	EmployeeID	Lastname
821	10944	6	Suyama
822	10956	6	Suyama
823	10959	6	Suyama
824	10965	6	Suyama
825	10973	6	Suyama
826	10999	6	Suyama
827	11019	6	Suyama
828	11025	6	Suyama
829	11031	6	Suyama
830	11045	6	Suyama
831	NULL	10	Urrutia

En el último registro se observa que en el campo OrderID tiene un valor NULL, lo cual indica que el empleado Urrutia no ha atendido ningún pedido

LEFT JOIN (IS NULL)

Muestra los registros de la tabla izquierda menos los registros coincidentes con la tabla derecha




```
SELECT <lista_campos>
                FROM <TablaA A>
LEFT JOIN  <TablaB B>
                 ON A.Key=B.Key
                                  WHERE B.Key IS
NULL
```

Ejemplo: Se desea conocer los empleados que no han atendido ningún pedido

La consulta SQL es:

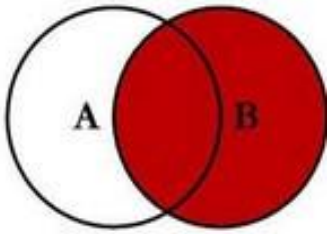
```
SELECT OrderID, E.EmployeeID, Lastname
FROM Employees E
LEFT JOIN Orders O
ON E.EmployeeID=O.EmployeeID
WHERE O.EmployeeID IS NULL
```

Se obtiene el siguiente resultado (un solo registro):

	OrderID	EmployeeID	Lastname
1	NULL	10	Umutia

RIGHT JOIN

Muestra los registros de la tabla derecha más los registros coincidentes con la tabla izquierda



SELECT <lista_campos> FROM <TablaA A>
RIGHT JOIN <TablaB B>
ON A.Key=B.Key

Ejemplo: Mostrar que productos ofrece cada proveedor independientemente si este lo hace o no

La consulta SQL es:

```
SELECT ProductName, CompanyName, ContactName
FROM Products P
RIGHT JOIN Suppliers S
ON P.SupplierID=S.SupplierID
```

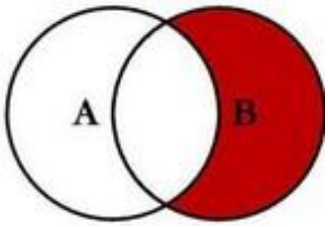
Se obtiene el siguiente resultado:

	ProductName	CompanyName	ContactName
74	Raclette Courdavault	Gai pâturage	Eliane Noz
75	Camembert Pierrot	Gai pâturage	Eliane Noz
76	Sirop d'érable	Forêts d'érables	Chantal Goulet
77	Tarte au sucre	Forêts d'érables	Chantal Goulet
78	NULL	Coca Cola	Iñaky Perez

En el último registro se verifica que el proveedor Coca Cola no ha ofrecido ningún producto

RIGHT JOIN (IS NULL)

Muestra los registros de la tabla derecha menos los registros coincidentes con la tabla izquierda



SELECT <lista_campos> FROM <TablaA A>
RIGHT JOIN <TablaB B>
ON A.Key=B.Key

WHERE A.Key IS NULL

Ejemplo: Mostrar que proveedor no ha ofrecido productos

La consulta SQL es:

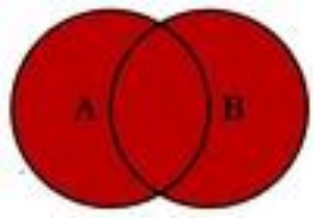
```
SELECT ProductName, CompanyName, ContactName
FROM Products P
RIGHT JOIN Suppliers S
ON P.SupplierID=S.SupplierID
WHERE P.SupplierID IS NULL
```

Se obtiene el siguiente resultado (un solo registro):

	ProductName	CompanyName	ContactName
1	NULL	Coca Cola	Iñaky Perez

FULL JOIN

Muestra los registros de la tabla izquierda y la tabla derecha más los registros coincidentes entre ambas




```
SELECT <lista_campos>
                    FROM <TablaA A>
FULL JOIN  <TablaB B> ON A.Key=B.Key
```

Ejemplo: En la siguiente consulta se muestra los productos que tengan o no asignado un proveedor y los proveedores independientemente si estos han ofrecido o no un producto

La consulta SQL es:

```
SELECT ProductName, CompanyName, ContactName
FROM Products P
FULL JOIN Suppliers S
ON P.SupplierID=S.SupplierID
```

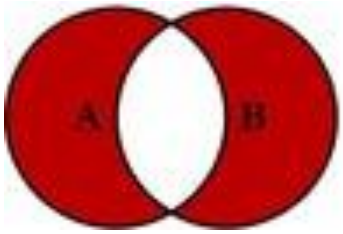
Se obtiene el siguiente resultado:

	ProductName	CompanyName	ContactName
1	Producto X	NULL	NULL
2	Chai	Exotic Liquids	Charlotte Cooper
3	Chang	Exotic Liquids	Charlotte Cooper
4	Aniseed Syrup	Exotic Liquids	Charlotte Cooper
	ProductName	CompanyName	ContactName
75	Raclette Courdavault	Gai pâturage	Eliane Noz
76	Camembert Pierrot	Gai pâturage	Eliane Noz
77	Sirop d'érable	Forêts d'érables	Chantal Goulet
78	Tarte au sucre	Forêts d'érables	Chantal Goulet
79	NULL	Coca Cola	Iñaky Perez

Observar que el primer registro indica que el Producto X no tiene proveedor asignado y en el último registro el Proveedor Coca Cola no ha ofrecido ningún producto.

FULL JOIN (IS NULL)

Muestra los registros de la tabla izquierda y la tabla derecha menos los registros coincidentes entre ambas




```
SELECT <lista_campos>
                FROM <TablaA A>
FULL JOIN  <TablaB B>
                ON A.Key=B.Key

WHERE A.Key IS NULL OR B.Key IS
NULL
```

Ejemplo: En la siguiente consulta se muestra los productos que no tienen asignado un proveedor y los proveedores que no han ofrecido un producto

La consulta SQL es:

```
SELECT ProductName, CompanyName, ContactName
FROM Products P
FULL JOIN Suppliers S
ON P.SupplierID=S.SupplierID
WHERE P.SupplierID IS NULL OR S.SupplierID IS NULL
```

Se obtiene el siguiente resultado (dos registros):

	ProductName	CompanyName	ContactName
1	Producto X	NULL	NULL
2	NULL	Coca Cola	Iñaky Perez

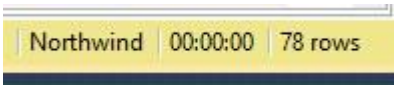
CROSS JOIN

Una combinación cruzada que no tenga una cláusula WHERE genera el producto cartesiano de las tablas involucradas en la combinación.
El tamaño del conjunto de resultados de un producto cartesiano es igual al número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla.

Ejemplo

Ejecutamos las siguientes consultas para conocer la cantidad de filas o registros tienen las siguientes tablas:

```
SELECT * FROM Products
```




```
SELECT * FROM Suppliers
```

Ahora ejecutamos la siguiente consulta:

```
SELECT ProductName, CompanyName, ContactName
FROM Products P
CROSS JOIN Suppliers S
```

Como resultado tenemos 2340 filas o registros, ya que si multiplicamos las 78 filas de la primera tabla por las 30 filas de la segunda obtenemos ese resultado

Sin embargo, si se agrega una cláusula WHERE, la combinación cruzada se comporta como una combinación interna (INNER JOIN)

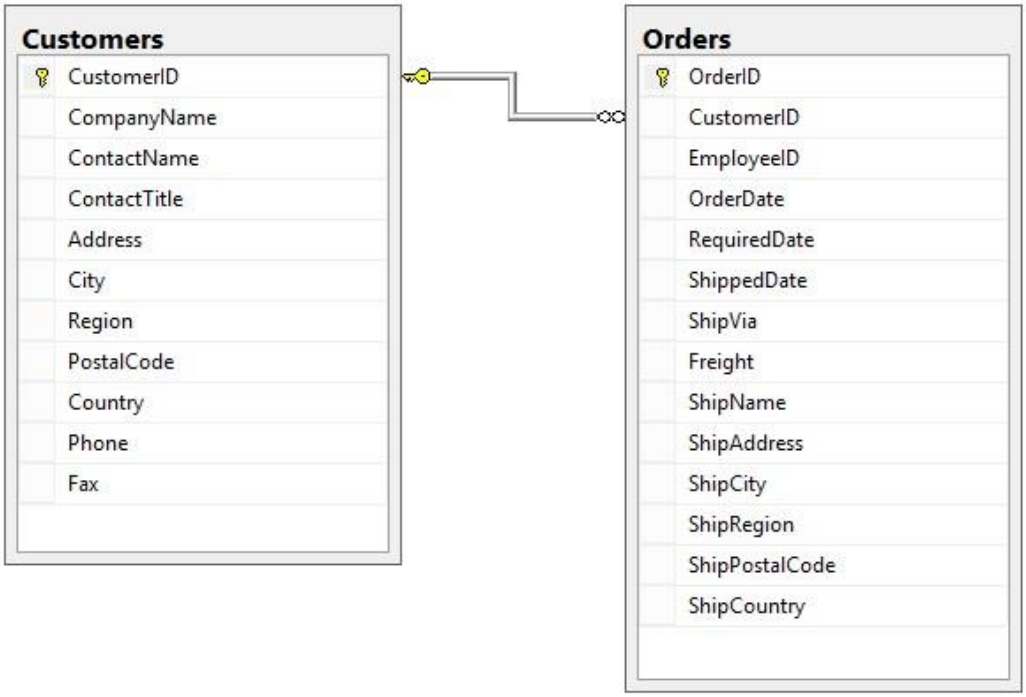
```
SELECT ProductName, CompanyName, ContactName
FROM Products P
CROSS JOIN Suppliers S
WHERE P.SupplierID=S.SupplierID
```

Obtenemos el mismo resultado al ejecutar la siguiente consulta:

```
SELECT ProductName, CompanyName, ContactName
FROM Products P
INNER JOIN Suppliers S
ON P.SupplierID=S.SupplierID
```

1. Haciendo uso de la base de datos **NORTHWIND o NORTHWND**

2. Se muestra la relación que existe entre las tablas **Customers y Orders**



3. Por medio de una instrucción INSERT agregar los siguientes datos a cada una de las tablas:

Tabla: **Customers**

	Campos	
	CustomerID	**CompanyNa**
1	TIPLE	Típicos Regior
2	FLOSU	Flores del Sur

Tabla: **Orders**

	Campo
	CustomerID
1	NULL

```
INSERT INTO Customers(CustomerID,CompanyName)
VALUES('TIPLE','Típicos Regionales'),
      ('FLOSU','Flores del Sur')

INSERT INTO Orders(CustomerID)
VALUES(NULL)
```

Ejercicio 1. Uso del INNER JOIN

2. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

3. Ejecutar la consulta para obtener los resultados
4. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
5. Ejecutar la consulta

Ejercicio 2. Uso del RIGHT JOIN

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
RIGHT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 3. Uso del RIGHT JOIN (IS NULL)

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
RIGHT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
WHERE Customers.CustomerID IS NULL
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 4. Uso del LEFT JOIN

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 5. Uso del LEFT JOIN (IS NULL)

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
WHERE Orders.CustomerID IS NULL
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 6. Uso del FULL JOIN

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
FULL JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 7. Uso del FULL JOIN (IS NULL)

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
FULL JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
WHERE Customers.CustomerID IS NULL OR Orders.CustomerID IS NULL
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando ALIAS para los nombres de las tablas
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio 8. Uso del CROSS JOIN

1. Digitar la siguiente consulta:

```
SELECT OrderID, Customers.CustomerID
FROM Customers
CROSS JOIN Orders
```

2. Ejecutar la consulta para obtener los resultados
3. Construir de nuevo la consulta implementando la cláusula WHERE
4. Ejecutar la consulta
5. Guardar los cambios

Ejercicio

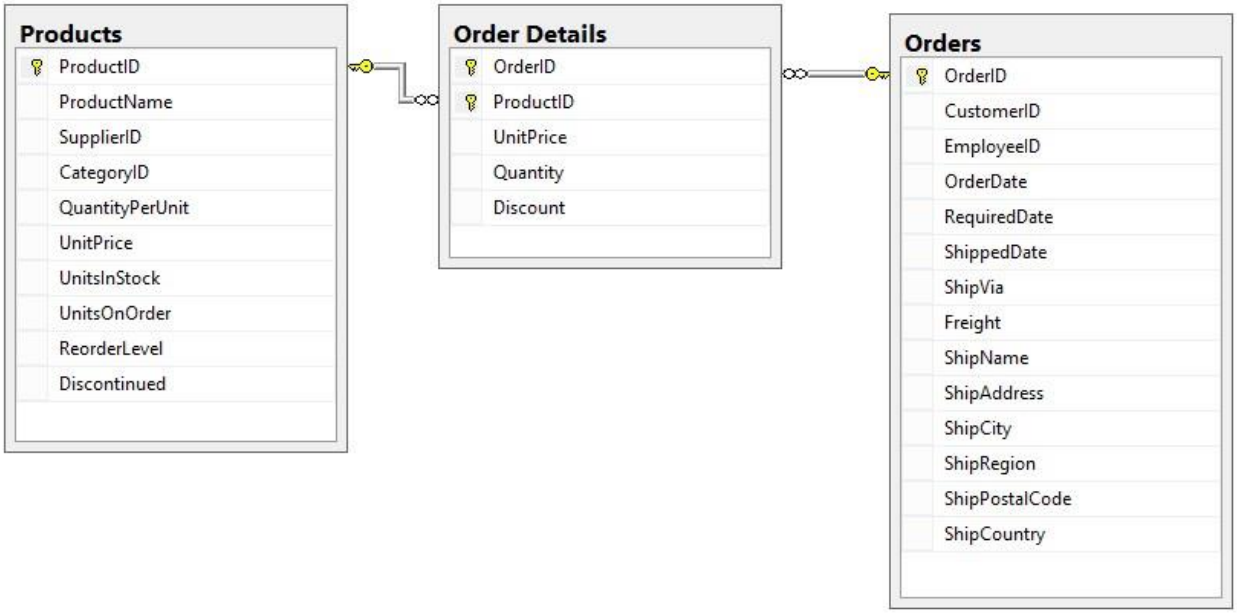
Usando la base de datos NORTHWIND o NORTHWND, resolver los siguientes ejercicios:

Ejercicio 1.

Haciendo uso de INNER JOIN mostrar los campos OrderDate y ProductID de las tablas Orders y Order Details donde el dato almacenado en el campo OrderDate sea igual 8 de Julio de 1996

Ejercicio 2.

Se desea mostrar las cantidades de cada producto que se han vendido y la fecha de la venta de cada uno de ellos, se debe tomar en cuenta el siguiente diagrama relacional:



Campos a mostrar: ProductID, ProductName, Quantity y OrderDate

Ejercicio 3:

Ingrese un nuevo registro a la tabla Employees únicamente para los campos LastName y FirtsName, el dato que Ud. quiera.

Dos nuevos registros a la tabla Orders únicamente para el campo EmployeeID, en el primer registro en el campo EmployeeID, debe agregar el código del nuevo empleado y en el segundo registro debe ingresar para ese campo un valor NULL

Crear consultas que implemente el uso de FULL, LEFT Y RIGHT JOIN

Campos a mostrar: LastName, EmployeeID de la tabla Employees

OrderDate de la tabla Orders

1. Creación de base de datos:
   Nombre de la base de datos: **Control_de_libros**

2. Crear las tablas tomando en cuenta:

   a. Crear las relaciones entre las tablas (llaves primarias y llaves foráneas).

   b. En las relaciones de las tablas implementar las instrucciones ON DELETE CASCADE y ON UPDATE CASCADE

   c. Deberá implementar también las restricciones (CHECK, UNIQUE y DEFAULT) en los campos de cada tabla que ud. cree necesarias.

3. El formato de las tablas es:

Tabla Autor:

CodigoAutor	PrimerNombre	PrimerApellido	FechaNacimiento	Nacionalidad	Edad
PL001	**Pablo**	**López**	**19/08/1960**	**Colombiana**	**54**
CM002	**Claudia**	**Martínez**	**10/06/1970**	**Salvadoreña**	**45**
PM003	**Patricio**	**Murry**	**12/12/1967**	**Española**	**47**
NH004	**Nuria**	**Hernández**	**03/09/1980**	**Colombiana**	**34**
HM005	**Helen**	**Martínez**	**22/11/1980**	**Española**	**34**
JR006	**José**	**Roldan**	**13/09/1967**	**Colombiano**	**54**

Tabla Editorial:

CodigoEditorial	**Nombre**	**Pais**
ED001	Omega 2000	Colombia
ED002	Anaya Multimedia	España
ED003	McGrawHill	Inglaterra
ED004	Reyes	México
ED005	Prentice Hall	Inglaterra

Tabla Libro:

CodigoLibro	Titulo	ISBN	AñoEdicion	CodigoEditorial
BDCOL00001	Fundamentos de Base de datos	12333-8999988	2004	ED001
BDESP00002	La Biblia de SQL Server 2008	3444-99888-88	2008	ED002
PRCOL00002	Programación orientada a objetos	8999-9999444	2011	ED001
DWING00003	Diseño Web y Hojas de estilo	300096-99999	2010	ED003
PRING00004	Programación en C/C++	45667-87878	2009	ED005
HJMEX00005	Uso de hojas de estilo con JavaScript	0990-87878787	2008	ED004
ABESP00006	Administración de Base de datos	585885-88484848	2010	ED002

Tabla Detalle_AutorLibro

CodigoAutor	CodigoLibro
PL001	BDCOL00001
NH004	BDCOL00001
CM002	PRCOL00002
PM003	BDESP00002
PM003	DWING00003
HM005	PRING00005
CM002	ABESP00006
NH004	HJMEX00005
JR006	DWING00003

4. Agregar los registros a las tablas

5. Crear las siguientes consultas:

a. Mostrar el primer nombre, primer apellido de los autores junto con el título de libro que estos han escrito

b. Mostrar el nombre de la editorial y el título del libro

c. Mostrar los títulos de los libros y el nombre de la editorial, donde esta sea del país de Inglaterra

d. Mostrar los nombres de los autores y el título del libro donde el año de edición sea el más actual

e. Mostrar los nombres de los autores y el título del libro donde el año de edición sea el menos actual

f. Agregue los datos necesarios a las tablas, para luego implementar las instrucciones LEFT JOIN, RIGHT JOIN Y FULL JOIN, como por ejemplo autores que no han escrito un libro todavía, editoriales que no han editado libros etc.