

# Consulta a múltiples tablas

---

29/03/2016

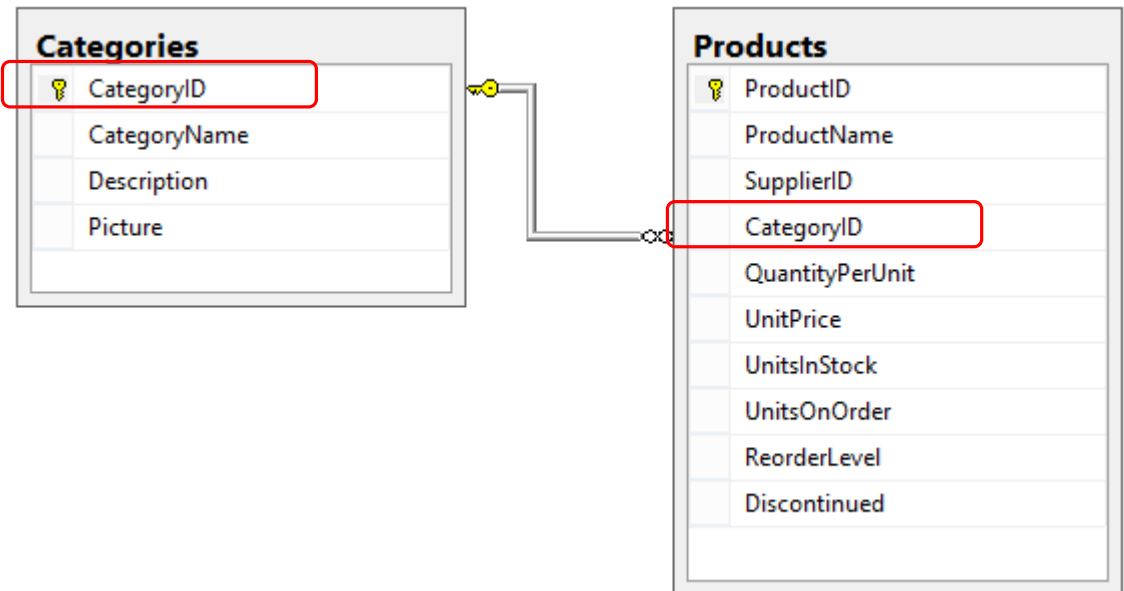
**Base de datos**  
**Grupo 01T**

# 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.

La combinación de campos de tablas distintas sólo es posible cuando se han definido campos relacionados entre tablas



# COMBINACIÓN DE 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.

# Uso de la instrucción JOIN

---

# SENTENCIA JOIN

---

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

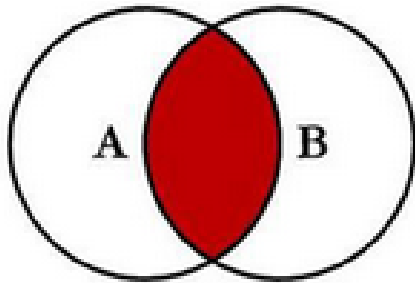
En el Lenguaje de Consultas Estructurado (SQL) hay tres tipos de JOIN: interno, externo y cruzado.

- 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

---



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

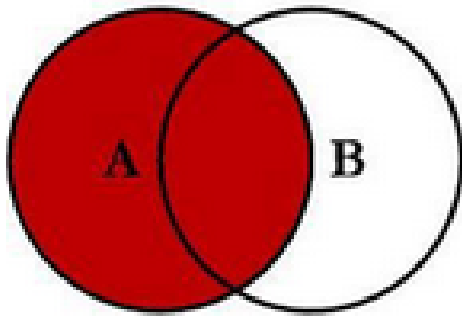
Ejemplo: Se desea conocer todos los productos que se encuentran en una orden

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

	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

# SQL JOINS

## LEFT JOIN



```
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

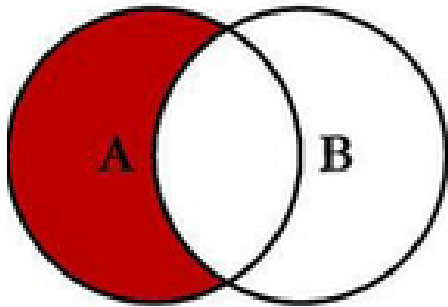
```
SELECT OrderID, E.EmployeeID, Lastname  
FROM Employees E  
LEFT JOIN Orders O  
ON E.EmployeeID=O.EmployeeID
```

	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	Umutia

# SQL JOINS

## LEFT JOIN (IS NULL)

---



```
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

```
SELECT OrderID, E.EmployeeID, Lastname
```

```
FROM Employees E
```

```
LEFT JOIN Orders O
```

```
ON E.EmployeeID=O.EmployeeID
```

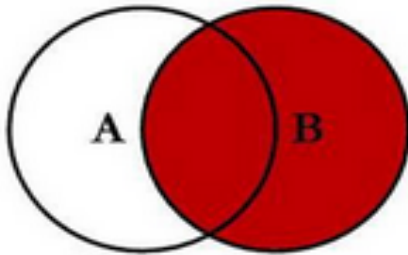
```
WHERE O.EmployeeID IS NULL
```

	OrderID	EmployeeID	Lastname
1	NULL	10	Umutia



# SQL JOINS

## RIGHT JOIN



```
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

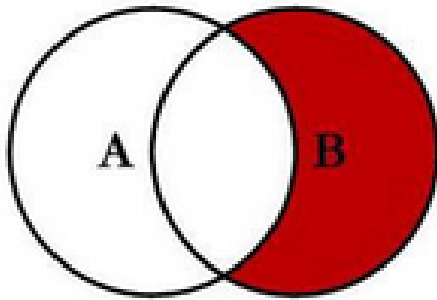
```
SELECT ProductName, CompanyName, ContactName  
FROM Products P  
RIGHT JOIN Suppliers S  
ON P.SupplierID=S.SupplierID
```

	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

# SQL JOINS

## RIGHT JOIN (IS NULL)

---



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

Ejemplo: Mostrar que proveedor no ha ofrecido productos

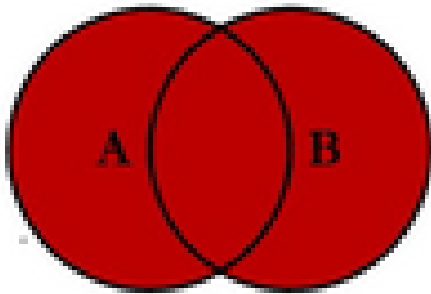
```
SELECT ProductName, CompanyName, ContactName  
FROM Products P  
RIGHT JOIN Suppliers S  
ON P.SupplierID=S.SupplierID  
WHERE P.SupplierID IS NULL
```

	ProductName	CompanyName	ContactName
1	NULL	Coca Cola	Iñaky Perez

# SQL JOINS

## FULL JOIN

---



```
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 hayan o no ofrecido un producto

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

# SQL JOINS

## FULL JOIN

Resultado de la consulta anterior

	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

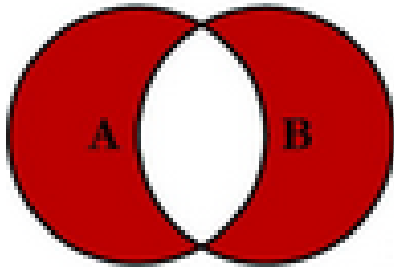
  

	ProductName	CompanyName	ContactName
75	Raclette Courdavault	Gai pâturage	Eliane Noz
76	Camembert Pierrot	Gai pâturage	Eliane Noz
77	Sirop d'érable	Forêts d'érables	Chantal Goulet
78	Tarte au sucre	Forêts d'érables	Chantal Goulet
79	NULL	Coca Cola	Iñaky Perez

# SQL JOINS

## FULL JOIN (IS NULL)

---



```
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

```
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
```

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

# SQL JOINS

## 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.

# SQL JOINS

## CROSS JOIN

---

### Ejemplo

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

```
SELECT * FROM Products
```

Northwind	00:00:00	78 rows
-----------	----------	---------

```
SELECT * FROM Suppliers
```

Northwind	00:00:00	30 rows
-----------	----------	---------

Ahora ejecutamos la siguiente consulta:

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

Northwind	00:00:00	2340 rows
-----------	----------	-----------

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

# SQL JOINS

## CROSS JOIN

---

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
```



# SubConsultas en SQL

---

# Subconsultas

---

Una SUBCONSULTA es una consulta T-SQL normal anidada dentro de otra consulta, se crean utilizando paréntesis en una instrucción SELECT que sirve como base para cualquier parte de los datos o de la condición de otra consulta.

Las SUBCONSULTAS devuelven grupos o arreglos de valores los cuales pueden servir de parámetros a otras búsquedas. Esto permite que los valores a buscar no sean estáticos sino dinámicos, y que la consulta no tenga que ser regenerada en vista de los cambios de información que puede haber en la BD.

# Subconsultas

---

Normalmente las SUBCONSULTAS se utilizan para satisfacer una o un par de las siguientes necesidades:

1. Desglosar una consulta en una serie de pasos lógicos.
2. Proporcionar un listado que va a ser el destino de una cláusula WHERE con [IN | EXISTS | ANY | ALL].
3. Proporcionar una búsqueda dirigida por cada registro individual de una consulta principal.

# Subconsultas

---

Una SUBCONSULTA puede devolver:

1. Una sola columna o un solo valor en cualquier lugar en donde pueda utilizarse una expresión de un sólo valor y puede compararse usando los siguientes operadores: =,<,>,<=,>=,<>,!> y !<.
2. Una sola columna o muchos valores que se pueden utilizar con el operador de comparación de listas IN en la cláusula WHERE.
3. Muchas filas que pueden utilizarse para comprobar la existencia, usando la palabra EXISTS en la cláusula WHERE.

# Subconsultas

---

**Sintaxis:**

**SELECT** <lista de selección>

**FROM** <Alguna Tabla>

**WHERE** <Alguna Columna> [**IN** | **EXISTS** | **ANY** | **ALL**]  
(**SELECT** <columna>

**FROM** <Alguna Tabla>

[**WHERE** <condición>]

)

# Subconsultas

---

## Ejemplo 1:

Mostrar los campos ProductID, ProductName y UnitPrice de todos los productos con cantidades mayores a 100

```
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE ProductID IN (SELECT productid
                    FROM [Order Details]
                    WHERE Quantity >= 100)
ORDER BY ProductName
```

	ProductID	ProductName	UnitPrice
1	17	Alice Mutton	1248.00
2	60	Camembert Pierrot	1088.00
3	2	Chang	729.60
4	39	Chartreuse verte	691.20
5	24	Guaraná Fantástica	172.80
6	44	Gula Malacca	622.40
7	10	Ikura	992.00
8	41	Jack's New England Clam Chowder	308.80
9	51	Manjimup Dried Apples	1696.00
10	55	Pâté chinois	762.00

# Subconsultas

---

## Ejemplo 2:

En este ejemplo se utiliza la instrucción EXISTS y obtenemos los mismos resultados de la consulta anterior:

Seleccionando un campo en específico en la subconsulta

```
SELECT ProductID, ProductName, UnitPrice
FROM Products
WHERE EXISTS (SELECT ProductID FROM [Order Details]
              WHERE Quantity >= 100
              AND Products.ProductID = [Order Details].ProductID)
ORDER BY ProductName
```

	ProductID	ProductName	UnitPrice
1	17	Alice Mutton	1248.00
2	60	Camembert Pierrot	1088.00
3	2	Chang	729.60
4	39	Chartreuse verte	691.20
5	24	Guaraná Fantástica	172.80
6	44	Gula Malacca	622.40
7	10	Ikura	992.00
8	41	Jack's New England Clam Chowder	308.80
9	51	Manjimup Dried Apples	1696.00
10	55	Pâté chinois	768.00

# Subconsultas

---

Seleccionando todos los campos en la subconsulta:

```
SELECT ProductID,ProductName,UnitPrice
FROM Products
WHERE EXISTS (SELECT * FROM [Order Details]
              WHERE Quantity>=100
              AND Products.ProductID=[Order Details].ProductID)
ORDER BY ProductName
```

	ProductID	ProductName	UnitPrice
1	17	Alice Mutton	1248.00
2	60	Camembert Pierrot	1088.00
3	2	Chang	729.60
4	39	Chartreuse verte	691.20
5	24	Guaraná Fantástica	172.80
6	44	Gula Malacca	622.40
7	10	Ikura	992.00
8	41	Jack's New England Clam Chowder	308.80
9	51	Manjimup Dried Apples	1696.00
10	55	Pâté chinois	768.00



# Subconsultas

---

## SUBCONSULTAS implementadas en la cláusula SELECT

Mostrar el máximo Precio unitario (UnitPrice) por cada pedido

```
SELECT O.OrderID, OrderDate, (SELECT MAX(UnitPrice)
FROM [Order Details] OD
WHERE OD.OrderID=O.OrderID) AS MaxPrecioUnitario
FROM Orders AS O
```

	OrderID	OrderDate	MaxPrecioUnitario
1	10248	1996-07-04 00:00:00.000	34.80

# Subconsultas

---

La consulta anterior se puede realizar también de la siguiente manera:

```
SELECT o.OrderID, OrderDate, MAX(od.UnitPrice) as MaxPrecioUnitario
FROM Orders o
inner join [Order Details] od
ON o.OrderID=od.OrderID
WHERE od.OrderID=10248
GROUP BY o.OrderID, OrderDate
```

	OrderID	OrderDate	MaxPrecioUnitario
1	10248	1996-07-04 00:00:00.000	34.80

# Subconsultas

---

## SUBCONSULTAS en las instrucciones UPDATE y DELETE

```
-- Actualiza la tabla Products, aumentando en un 20% el precio unitario (UnitPrice)  
-- del producto, donde el nombre de la categoria comience con la letra B
```

```
UPDATE Products SET UnitPrice=UnitPrice*1.2  
WHERE CategoryID IN (SELECT CategoryID FROM Categories  
                     WHERE CategoryName LIKE 'B%')
```

# Subconsultas

---

## SUBCONSULTAS en las instrucciones UPDATE y DELETE

En la siguiente consulta se elimina el o los clientes cuya fecha de pedido sea mayor a 2014-01-01

```
DELETE Customers WHERE  
CustomerID IN (SELECT CustomerID FROM Orders  
                WHERE OrderDate > '2014-01-01')
```