

Instrucciones SQL

Laboratorio Bases de Datos I

CONTENIDOS

COMBINACIONES

- ☐ Uniones de tablas.
- ☐ Tipos de uniones:
 - CROSS
 - INNER
 - OUTER
- ☐ Subconsultas.

Combinaciones - JOIN

Las combinaciones permiten recuperar datos de dos o más tablas según las relaciones lógicas entre ellas.



Las combinaciones indican al motor de base de datos cómo debe usar los datos de una tabla para seleccionar las filas de otra tabla.

Combinaciones – JOIN

Condiciones de combinación

❑ **FROM** *first_table* **join_type** *second_table* [**ON** (*join_condition*)]

```
SELECT ProductID, Purchasing.Vendor.VendorID, Name
FROM Purchasing.ProductVendor
INNER JOIN Purchasing.Vendor ON
(Purchasing.ProductVendor.VendorID =
Purchasing.Vendor.VendorID)
WHERE StandardPrice > $10 AND Name LIKE N'F%'
```

Especificar las condiciones de la combinación en la cláusula **FROM** ayuda a separarlas de cualquier otra condición de búsqueda que se pueda especificar en una cláusula **WHERE**.

Combinaciones – JOIN

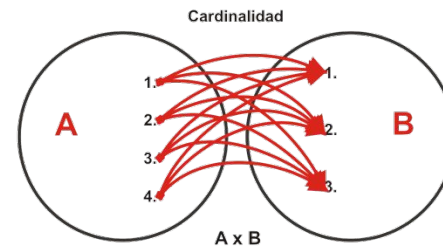
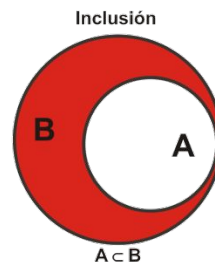
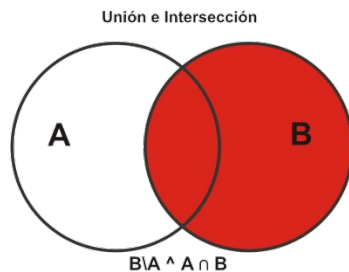
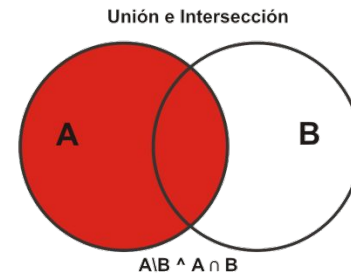
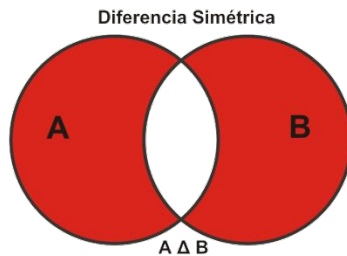
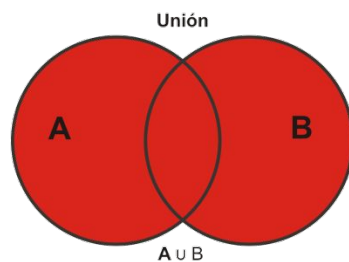
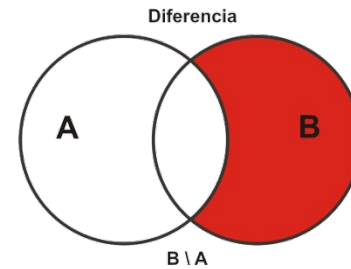
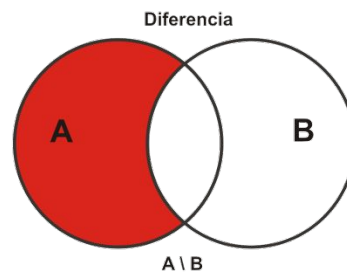
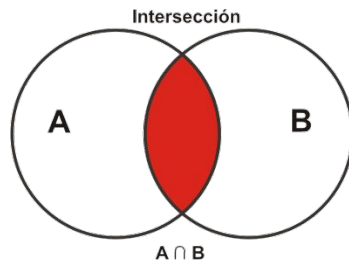
Condiciones de combinación

Condición de combinación especificada en la cláusula WHERE
(**no recomendado**)

```
SELECT pv.ProductID, v.VendorID, v.Name  
FROM Purchasing.ProductVendor pv, Purchasing.Vendor v  
WHERE pv.VendorID = v.VendorID AND StandardPrice > $10  
      AND Name LIKE N'F%'
```

Combinaciones – JOIN

Tipos de combinaciones

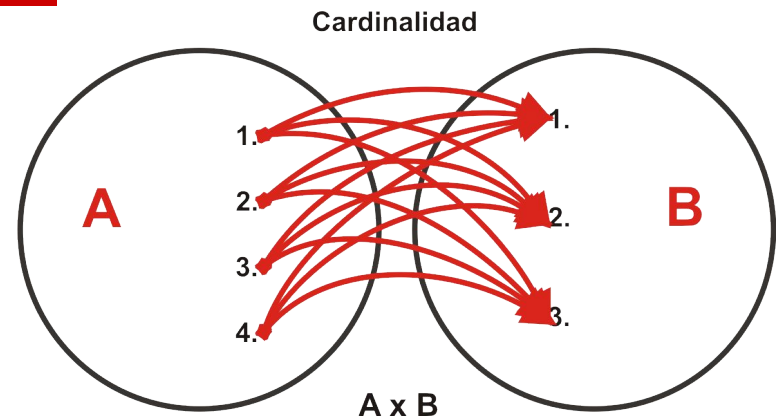


Combinaciones – JOIN

CROSS JOIN – Producto Cartesiano

Genera el producto cartesiano de las tablas involucradas en la combinación.

El resultado es igual al número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla.



--- ANSI

SELECT count(*)

FROM Sales.SalesPerson p **CROSS JOIN** Sales.SalesTerritory t

--no ANSI

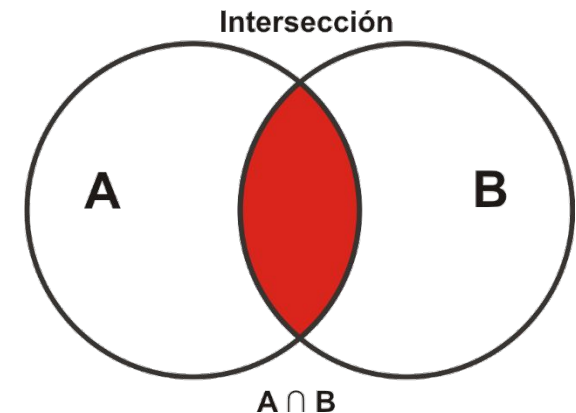
SELECT count(*)

FROM Sales.SalesPerson p, Sales.SalesTerritory t

Combinaciones – JOIN

INNER JOIN – Combinación Interna

Los valores de las columnas que se están combinando se comparan mediante un operador de comparación.



```
--- ANSI
SELECT *
FROM HumanResources.Employee AS e
INNER JOIN Person.Contact AS c
ON e.ContactID = c.ContactID
```

```
--- no ANSI
SELECT *
FROM HumanResources.Employee AS e, Person.Contact AS c
WHERE e.ContactID = c.ContactID
```


Combinaciones – JOIN

INNER JOIN – Combinación Interna

Combinar mas de dos tablas

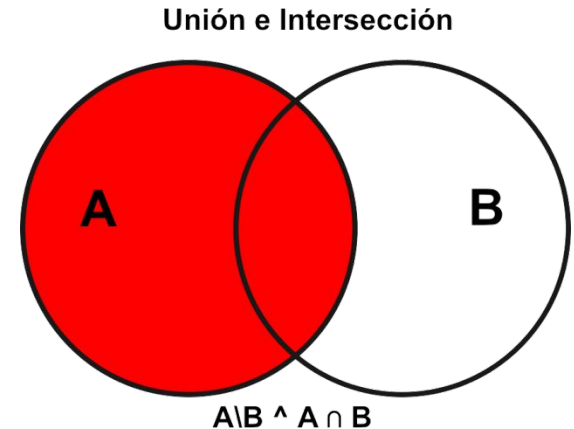
Las cláusulas **FROM** pueden contener varias especificaciones de combinaciones. Esto permite que se combinen muchas tablas en una consulta única.

```
SELECT p.Name, v.Name  
FROM Production.Product p  
JOIN Purchasing.ProductVendor pv ON p.ProductID = pv.ProductID  
JOIN Purchasing.Vendor v ON pv.VendorID = v.VendorID  
WHERE ProductSubcategoryID = 15  
ORDER BY v.Name
```

Combinaciones – JOIN

LEFT OUTER JOIN – Combinación Externa

Incluye en el resultado todas las filas de la tabla 'A' (izquierda), tanto si hay una coincidencia en la tabla 'B' como si no la hay.



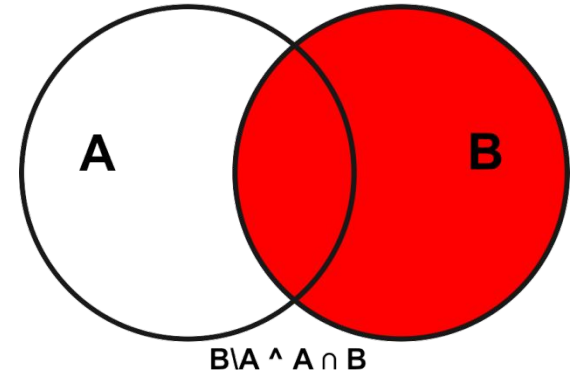
```
SELECT p.Name, pr.ProductReviewID
FROM Production.Product p
LEFT OUTER JOIN Production.ProductReview pr
ON p.ProductID = pr.ProductID
```

Combinaciones – JOIN

RIGHT OUTER JOIN – Combinación Externa

Unión e Intersección

Incluye en el resultado todas las filas de la tabla 'B' (derecha), tanto si hay una coincidencia en la tabla 'A' como si no la hay.



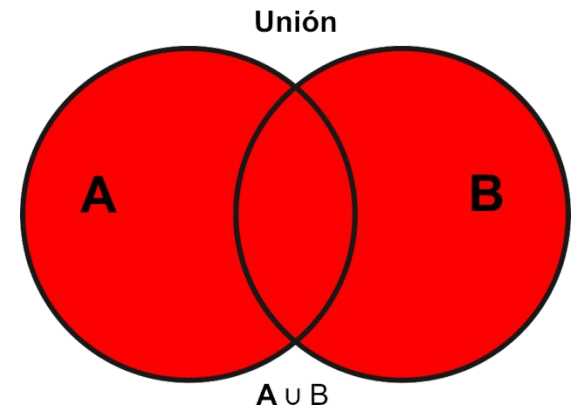
```
SELECT st.Name AS Territory, sp.SalesPersonID  
FROM Sales.SalesTerritory st  
RIGHT OUTER JOIN Sales.SalesPerson sp  
ON st.TerritoryID = sp.TerritoryID ;
```

```
WHERE st.SalesYTD < $2000000;
```

Combinaciones – JOIN

FULL OUTER JOIN – Combinación Externa

Incluye todas las filas de ambas tablas, con independencia de que la otra tabla tenga o no un valor coincidente.



```
SELECT th.ProductID, pph.ProductID
FROM Production.TransactionHistory th
FULL OUTER JOIN Production.ProductListPriceHistory pph
ON th.ProductID = pph.ProductID
```

Combinaciones – JOIN

AUTOCOMBINACIONES

Se utiliza para unir una tabla consigo misma, comparando valores de dos columnas con el mismo tipo de datos.

```
SELECT DISTINCT pv1.ProductID, pv1.VendorID FROM  
Purchasing.ProductVendor pv1  
INNER JOIN Purchasing.ProductVendor pv2  
ON pv1.ProductID = pv2.ProductID AND pv1.VendorID <>  
pv2.VendorID  
ORDER BY pv1.ProductID
```

Subconsultas

Una subconsulta es una consulta anidada en una instrucción SELECT, INSERT, UPDATE o DELETE, o bien en otra subconsulta.

```
SELECT Name
FROM Production.Product
WHERE ListPrice =
    (SELECT ListPrice
     FROM Production.Product
     WHERE Name = 'Chainring Bolts' )
```

Igual sintaxis que otra sentencia SELECT.
Aparece encerrada entre paréntesis.

Subconsultas

Muchas de las instrucciones que incluyen subconsultas se pueden formular también como combinaciones. Otras preguntas se pueden formular sólo con subconsultas.

```
SELECT Prd1. Name  
FROM Production.Product AS Prd1  
INNER JOIN Production.Product AS Prd2  
ON Prd1.ListPrice = Prd2.ListPrice  
WHERE Prd2. Name = 'Chainring Bolts'
```

Subconsultas

Las subconsultas se pueden utilizar en cualquier parte en la que se permita una expresión

```
SELECT Ord.SalesOrderID, Ord.OrderDate,  
      (SELECT MAX(OrdDet.UnitPrice)  
      FROM AdventureWorks.Sales.SalesOrderDetail AS OrdDet  
      WHERE Ord.SalesOrderID = OrdDet.SalesOrderID) AS  
      MaxUnitPrice  
FROM AdventureWorks.Sales.SalesOrderHeader AS Ord
```


Subconsultas

Una subconsulta anidada en la instrucción **externa SELECT** es una consulta **SELECT** normal, que incluye los componentes normales de la lista de selección, con una cláusula normal FROM, y una cláusula opcional WHERE | GROUP BY | HAVING.

La **consulta SELECT de una subconsulta** se presenta siempre entre paréntesis y sólo puede incluir una cláusula ORDER BY cuando se especifica también una cláusula TOP. Se puede disponer de hasta 32 niveles de anidamiento.

Subconsultas

Las instrucciones que incluyen una subconsulta normalmente tienen uno de estos formatos:

- ❑ **WHERE** expression [NOT] IN (subquery)
- ❑ **WHERE** expression comparison_operator [ANY | ALL] (subquery)
- ❑ **WHERE** [NOT] EXISTS (subquery)

Subconsultas

Hay tres tipos básicos de subconsultas:

- Las que **operan en listas** especificadas con **IN** o modificadas por un operador de comparación mediante **ANY** o **ALL**.
- Las que se especifican con un **operador de comparación** sin modificar y deben devolver un solo valor.
- Las que son **pruebas de existencia** especificadas con **EXISTS**.

Subconsultas

Listas

- El resultado de una subconsulta especificada con **IN** (o con **NOT IN**) es una lista de cero o más valores. Una vez que la consulta devuelve los resultados, la consulta externa hace uso de ellos.

```
SELECT SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
WHERE SalesOrderID IN (
    SELECT SalesOrderID
    FROM Sales.SalesOrderDetail
    GROUP BY SalesOrderID
    HAVING count (*) > 70)
```

Subconsultas

Operadores de comparación modificados

- Los operadores de comparación que presentan una subconsulta se pueden modificar mediante las palabras clave **ALL** o **ANY**. Las subconsultas presentadas con un operador de comparación modificado devuelven una lista de cero o más valores.

```
SELECT SalesOrderID, OrderDate, SubTotal
FROM Sales.SalesOrderHeader
WHERE SubTotal >ALL (
    SELECT TOP 5 SUM(OrderQty * UnitPrice) AS TotAmount
    FROM Sales.SalesOrderDetail
    GROUP BY SalesOrderID
    ORDER BY TotAmount DESC)
```

Subconsultas

Operadores de comparación

- Las subconsultas se pueden presentar con uno de los operadores de comparación (=, < >, >, > =, <, ! >, ! <, or < =).
- Una subconsulta precedida de un operador de comparación debe devolver un valor individual en lugar de una lista de valores. **Se debe estar suficientemente familiarizado con los datos y con la naturaleza del problema para saber que la subconsulta devolverá exactamente un valor.**

```
SELECT Name FROM Production.Product  
WHERE ListPrice > (  
    SELECT AVG (ListPrice)  
    FROM Production.Product)
```

Subconsultas

[NOT] EXISTS

- Funciona como una prueba de existencia. La cláusula WHERE de la consulta externa comprueba si existen las filas devueltas por la subconsulta. En realidad, la subconsulta **no produce ningún dato**, devuelve el valor **TRUE** o **FALSE**.
 - La palabra clave EXISTS no viene precedida de un nombre de columna, constante u otra expresión.
 - La lista de selección de una subconsulta que se especifica con EXISTS casi siempre consta de un asterisco (*)

Subconsultas

[NOT] EXISTS

```
USE AdventureWorks2008R2;
GO
SELECT Name
FROM Production.Product
WHERE EXISTS
    (SELECT *
     FROM Production.ProductSubcategory
     WHERE ProductSubcategoryID =
         Production.Product.ProductSubcategoryID
     AND Name = 'Wheels')
```


Subconsultas

Reglas de las subconsultas (Algunas restricciones)

- ❑ La lista de selección de una subconsulta que se especifica con un operador de comparación, sólo puede incluir un nombre de expresión o columna (excepto EXISTS e IN, que operan en SELECT * o en una lista respectivamente).
- ❑ Si la cláusula WHERE de una consulta externa incluye un nombre de columna, debe ser compatible con una combinación con la columna indicada en la lista de selección de la subconsulta.
- ❑ Puesto que deben devolver un solo valor, las subconsultas que se especifican con un operador de comparación sin modificar (no seguido de la palabra clave ANY o ALL) no pueden incluir las cláusulas GROUP BY y HAVING.
- ❑ La palabra clave DISTINCT no se puede usar con subconsultas que incluyan GROUP BY.
- ❑ Sólo se puede especificar ORDER BY si se especifica también TOP.
- ❑ Una vista creada con una subconsulta no se puede actualizar.
- ❑ La lista de selección de una subconsulta especificada con EXISTS, por convención, tiene un asterisco (*) en lugar de un solo nombre de columna. Las reglas de una subconsulta especificada con EXISTS son idénticas a las de una lista de selección.

Fin Tema