

5ta. Parte

Bases de Datos Relacionales

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

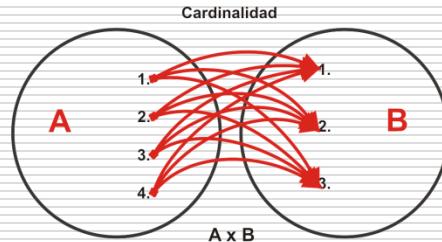
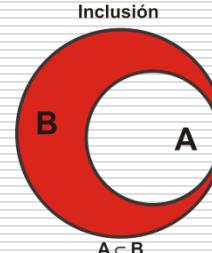
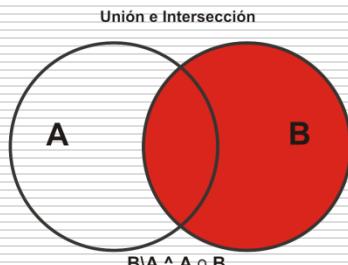
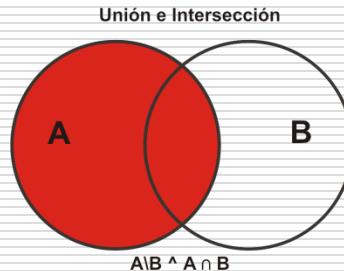
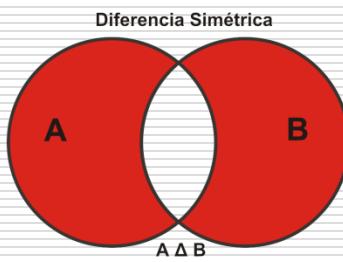
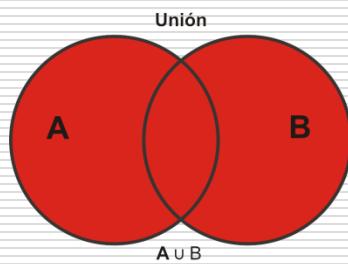
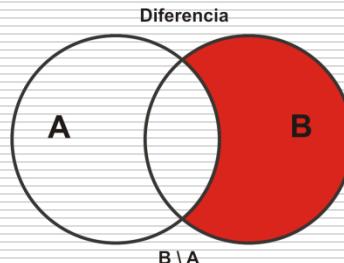
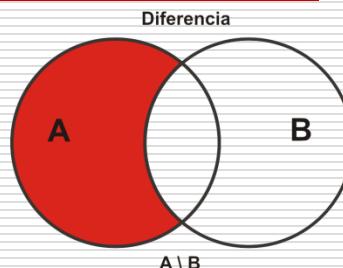
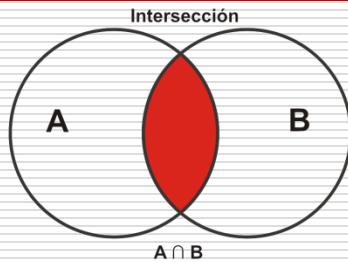
Condiciones de combinación

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

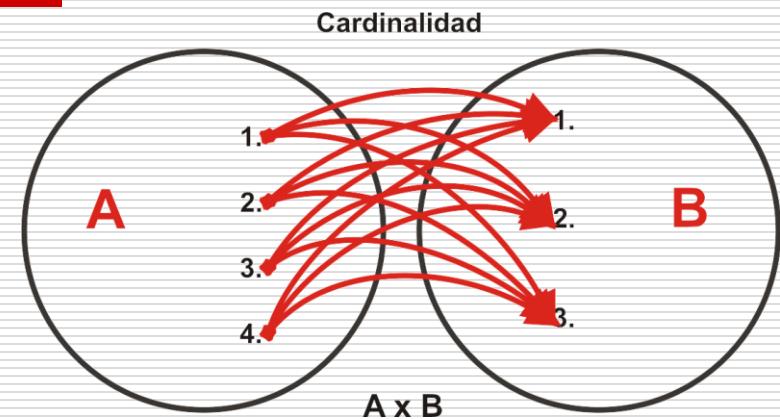


Combinaciones – JOIN

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
```

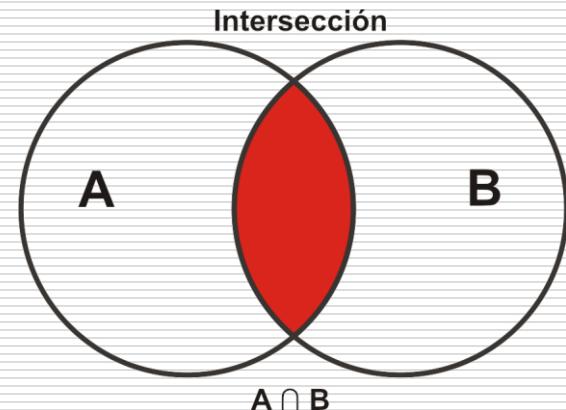
--no ANSI

```
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.

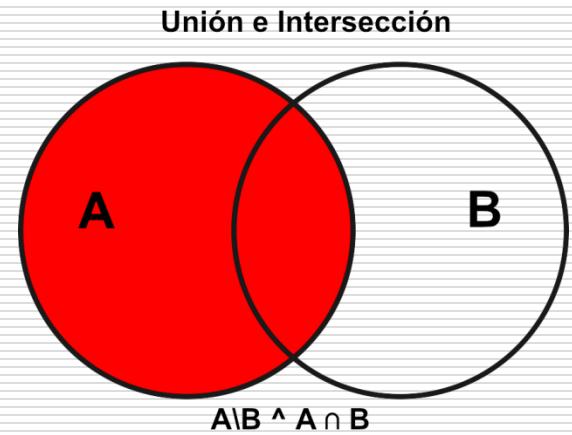
```
SELECT p.Name, v.Name  
FROM Production.Product p  
JOIN Purchasing.ProductVendor pv ON p.ProductID = pv.ProductID  
JOIN Purchasing.Vendor v ON pv.VendorID = v.VendorID  
WHERE ProductSubcategoryID = 15  
ORDER BY v.Name
```

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.



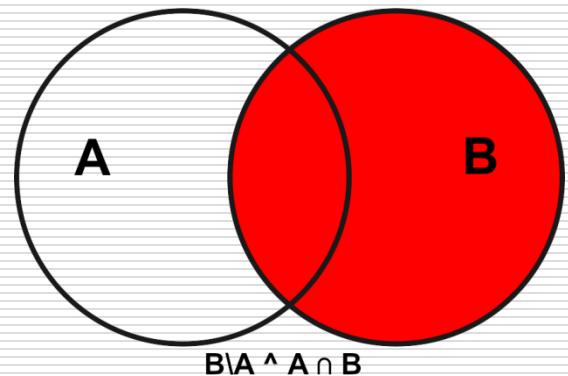
```
SELECT p.Name, pr.ProductReviewID  
FROM Production.Product p  
LEFT OUTER JOIN Production.ProductReview pr  
ON p.ProductID = pr.ProductID
```

Combinaciones – JOIN

RIGHT OUTER JOIN - Combinación Externa

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.

Unión e Intersección



```
SELECT st.Name AS Territory, sp.SalesPersonID
```

```
FROM Sales.SalesTerritory st
```

```
RIGHT OUTER JOIN Sales.SalesPerson sp
```

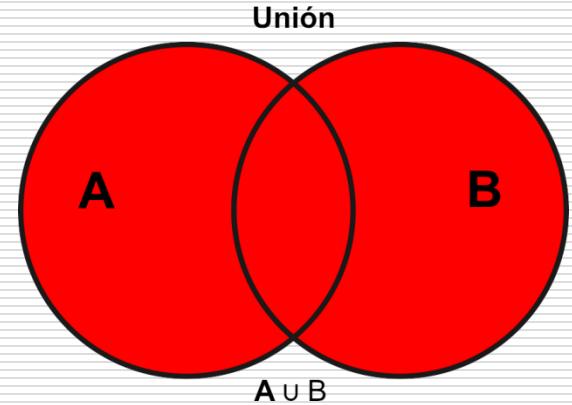
```
ON st.TerritoryID = sp.TerritoryID ;
```

```
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

- Qué son las subconsultas**
- Condiciones**
- Tipos de subconsultas**
- Ejercicio práctico**

Subconsultas. Condiciones

- La consulta SELECT de una subconsulta se presenta siempre entre paréntesis.
- Sólo puede incluir una cláusula ORDER BY cuando se especifica también una cláusula TOP.
- 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. (Coincidencia de tipos de datos).
- Utilizar operadores a nivel de fila para subconsultas que devuelvan solo una fila.
- Utilizar operadores que actúan sobre varios registros para subconsultas que devuelven más de una fila.
- Muchas de las instrucciones que incluyen subconsultas se pueden formular también como combinaciones.

Subconsultas. Tipos

- Subconsultas con operadores de comparación (mono-registro)
- Subconsultas con operadores de conjuntos (multi-registro)
- Subconsultas correlacionadas con EXISTS
- Subconsultas en cláusula FROM (tablas derivadas)

Subconsultas. Mono-Registro

- **Subconsultas con operadores de comparación**
(mono-registro)
- La subconsulta devuelve un único registro.
- Se utilizan operadores de comparación (**=, < >, >, > =, <, ! >, ! <, or < =**)

```
SELECT ...
FROM Tabla
WHERE columna = (
    SELECT MAX(columna)
    FROM tabla);
```

Subconsultas. Multi-Registros

- 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.
- Se utilizan operadores como [NOT] IN, <>=ANY, <>=ALL
- No pueden evaluar NULLs, por lo que las filas son siempre falsas.

```
SELECT ...
FROM Tabla
WHERE columna in ( SELECT columna
                      FROM tabla);
```

Subconsultas. Exists

- Funciona como una prueba de existencia
- La cláusula WHERE de la consulta externa comprueba si existen las filas devueltas por la subconsulta. La subconsulta no produce ningún dato, devuelve el valor TRUE o FALSE.
- Se utiliza el operador [NOT] EXISTS. Sintaxis: *WHERE [NOT] EXISTS (subquery)*
- EXISTS puede evaluar NULL, por lo que las filas se pueden devolver.

```
SELECT columna/s  
FROM tabla t1  
WHERE EXISTS (  
    SELECT *  
    FROM tabla t2  
    WHERE t1.columna = t2.columna)
```

Subconsultas. Tablas Derivadas

- Se llaman tablas derivadas a las subconsultas cuando se les da un nombre y son usadas después de la cláusula FROM.
- Generalmente se utiliza cuando se quiere anidar funciones agregadas.
- Sintaxis: *SELECT * FROM (SELECT * FROM tabla) AS ablaDerivada;*

```
SELECT AVG(resultado.cantidad) promedio  
FROM (  
    SELECT COUNT(*) cantidad  
    FROM consorcio c  
    GROUP BY c.idzona  
 ) AS resultado;
```

Subconsultas

Ej. "Mostrar todos los consorcios que pertenezcan a la zona que mas consorcios agrupa."

```
SELECT *  
FROM consorcio  
WHERE idzona = ( SUBCONSULTA QUE DEVUELVA EL IDZONA DE  
LA QUE MAS CONSORCIO TENGA ).....
```

Subconsulta:

```
SELECT TOP 1 idzona, COUNT(*) cantidad  
FROM consorcio  
GROUP BY idzona ORDER BY count(*) DESC
```

Subconsultas

Unir ambas consultas. Es posible ?

```
SELECT *
FROM consorcio
WHERE idzona = (SELECT TOP 1 idzona, COUNT(*) cantidad
FROM consorcio
GROUP BY idzona ORDER BY count(*) DESC )
```

Como lo solucionamos ?

```
SELECT *
FROM consorcio
WHERE idzona = (SELECT TOP 1 idzona cantidad
FROM consorcio
GROUP BY idzona ORDER BY count(*) DESC )
```

Subconsultas

Otra opción. Con tablas derivadas (Un tipo de subconsulta).

```
SELECT *
FROM consorcio
WHERE idzona = (SELECT idzona
FROM (SELECT TOP 1 idzona cantidad
FROM consorcio
GROUP BY idzona ORDER BY count(*) DESC ) aux)
```

Bibliografia

Libros en pantalla SQL Server (T-SQL)

[https://technet.microsoft.com/es-es/library/ms189575\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms189575(v=sql.105).aspx)

Libros en pantalla MySQL

<https://dev.mysql.com/doc/refman/5.7/en/subqueries.html>

Database Administration - ANSI SQL Standards and Guidelines

<http://www.whoishostingthis.com/resources/ansi-sql-standards/>

Date, C.J. *Introducción a los sistemas de bases de datos*. 7ma Ed. Pearson Education, México, 2001. ISBN: 968-444-419-2

Hotek, M. *Microsoft SQL Server 2008 Step by Step*. Microsoft Press, Redmond, Washington 2009.



Fin Tema

