

SQL para Análisis de Datos

Módulo V



Temario

- SQL – Joins
 - Fundamentos
 - Inner Join
 - Outer Joins
 - LEFT JOIN o LEFT OUTER JOIN
 - RIGHT JOIN o RIGHT OUTER JOIN
 - FULL JOIN o FULL OUTER JOIN
 - Cross Join

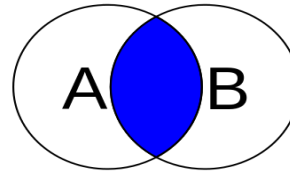
SQL

Joins

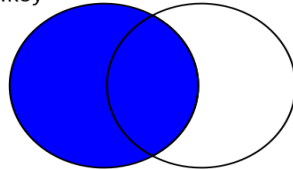
Fundamentos

- Mediante el uso de Joins, se pueden recuperar los datos de dos o más tablas en función de las relaciones lógicas entre las mismas
- Al utilizar los distintos tipos de Joins se le indica al motor de base de datos cómo utilizar los datos de una tabla para seleccionar las filas de otra tabla
- Se debe especificar la condición de combinación entre las tablas:
 - Indicando las columnas que se utilizan para la unión
 - Especificando el operador lógico de comparación a utilizar (por ejemplo, = o <>)

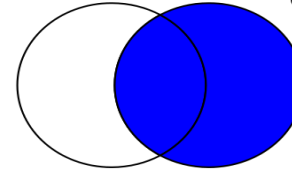
```
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key
```



```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
```

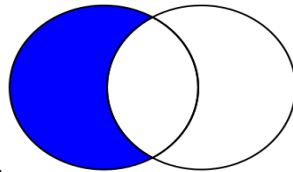


```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```

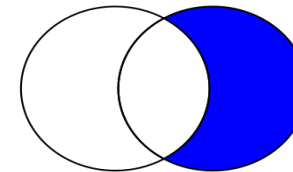


SQL JOINS

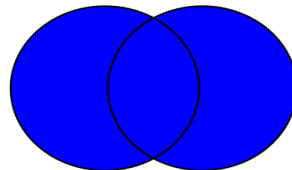
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```



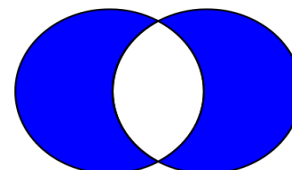
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
```



```
SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL
```



This work is licensed under a Creative Commons Attribution 3.0 Unported License.
 Author: <http://commons.wikimedia.org/wiki/User:Arbeck>

SQL

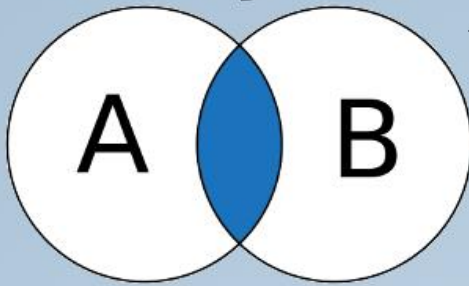
Joins

Inner Join

INNER JOIN o JOIN

- Es utilizado principalmente para obtener los datos de 2 tablas relacionadas donde cada registro de A le corresponde algún registro de B y viceversa
- INNER JOIN o JOIN son equivalentes

INNER JOIN



```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```

SQL

Joins

Inner Join

INNER JOIN o JOIN

- Ejemplo:

```
SELECT p.FirstName, p.LastName, e.LoginID
FROM HumanResources.Employee AS e
INNER JOIN Person.Person AS p
ON e.BusinessEntityID = p.BusinessEntityID
ORDER BY p.LastName
```

	FirstName	LastName	LoginID
1	Syed	Abbas	adventure-works\syed0
2	Kim	Abercrombie	adventure-works\kim1
3	Hazem	Abolrous	adventure-works\hazem0
4	Pilar	Ackeman	adventure-works\pilar0
5	Jay	Adams	adventure-works\jay0
6	François	Ajenstat	adventure-works\françois0
7	Amy	Alberts	adventure-works\amy0

En este ejemplo se obtienen los nombre, apellidos y loginID de todos los empleados

INNER JOIN o JOIN

- Ejemplo:

```
SELECT soh.SalesOrderID, soh.DueDate, sod.OrderQty
FROM Sales.SalesOrderHeader AS soh
JOIN Sales.SalesOrderDetail AS sod
ON soh.SalesOrderID = sod.SalesOrderID
WHERE sod.ProductID = 718
ORDER by sod.OrderQty DESC
```

	SalesOrderID	DueDate	OrderQty
1	47395	2002-09-13 00:00:00.000	7
2	53472	2003-09-13 00:00:00.000	7
3	47027	2002-08-13 00:00:00.000	6
4	46660	2002-07-13 00:00:00.000	6
5	46671	2002-07-13 00:00:00.000	5
6	47369	2002-09-13 00:00:00.000	5
7	51123	2003-07-13 00:00:00.000	5

En este ejemplo se obtienen las órdenes (número y fecha) de todas las órdenes dónde se pidió el producto con ID 718. Además, se obtienen las cantidades ordenadas del mismo.

INNER JOIN o JOIN

- Ejemplo:

```
SELECT DISTINCT p.ProductID, p.Name, p.ListPrice, sd.UnitPrice AS  
'Selling Price'  
FROM Sales.SalesOrderDetail AS sd  
JOIN Production.Product AS p  
ON sd.ProductID = p.ProductID AND sd.UnitPrice < p.ListPrice  
WHERE p.ProductID = 718
```

	ProductID	Name	ListPrice	Selling Price
1	718	HL Road Frame - Red, 44	1431,50	758,0759
2	718	HL Road Frame - Red, 44	1431,50	780,8182
3	718	HL Road Frame - Red, 44	1431,50	858,90

En este ejemplo se obtienen las variaciones de precio del producto con ID 718, donde el mismo fue vendido a un precio menor a su precio de lista

SQL

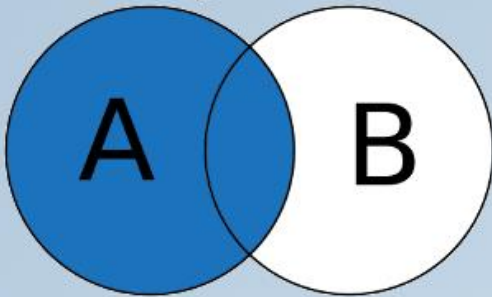
Joins

Outer Join

LEFT JOIN o LEFT OUTER JOIN

- Es utilizado para obtener todos los registros de la tabla A (izquierda o primera) y los registros relacionados de la tabla B

LEFT JOIN



```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```

SQL

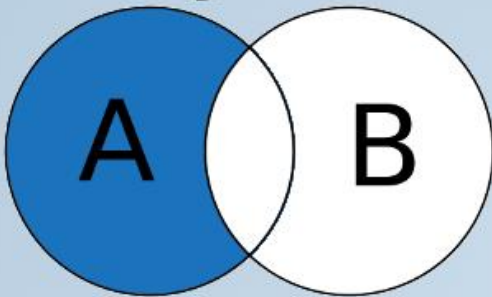
Joins

Outer Join

LEFT JOIN o LEFT OUTER JOIN

- Se puede condicionar para obtener los registros de la tabla A que no se relacionan con la tabla B

LEFT JOIN



```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

SQL

Joins

Outer Join

LEFT JOIN o LEFT OUTER JOIN

- Ejemplo:

```
SELECT p.Name, pr.ProductReviewID, pr.ReviewerName, pr.Comments
FROM Production.Product p
LEFT OUTER JOIN Production.ProductReview pr
ON p.ProductID = pr.ProductID
ORDER BY pr.ProductReviewID DESC
```

	Name	ProductReviewID	ReviewerName	Comments
1	Road-550-W Yellow, 40	4	Laura Norman	The Road-550-W from Adventure Works C
2	HL Mountain Pedal	3	Jill	Maybe it's just because I'm new to mounta
3	HL Mountain Pedal	2	David	A little on the heavy side, but overall the er
4	Mountain Bike Socks, M	1	John Smith	I can't believe I'm singing the praises of a p
5	Mountain Bike Socks, L	NULL	NULL	NULL
6	Sport-100 Helmet, Blue	NULL	NULL	NULL
7	AWC Logo Cap	NULL	NULL	NULL

AdventureWorks2008 | 00:00:00 | 505 rows

En este ejemplo se obtienen **todos los productos** (tengan comentarios o no) y los comentarios de los productos que correspondan

SQL

Joins

Outer Join

LEFT JOIN o LEFT OUTER JOIN

- Ejemplo:

```
SELECT p.Name, pr.ProductReviewID, pr.ReviewerName, pr.Comments
FROM Production.Product p
LEFT OUTER JOIN Production.ProductReview pr
ON p.ProductID = pr.ProductID
WHERE pr.ProductID IS NULL
```

	Name	ProductReviewID	ReviewerName	Comments
1	Adjustable Race	NULL	NULL	NULL
2	Bearing Ball	NULL	NULL	NULL
3	BB Ball Bearing	NULL	NULL	NULL
4	Headset Ball Bearings	NULL	NULL	NULL
5	Blade	NULL	NULL	NULL

AdventureWorks2008 | 00:00:00 | 501 rows

En este ejemplo se obtienen todos los productos que no tuvieron comentarios

SQL

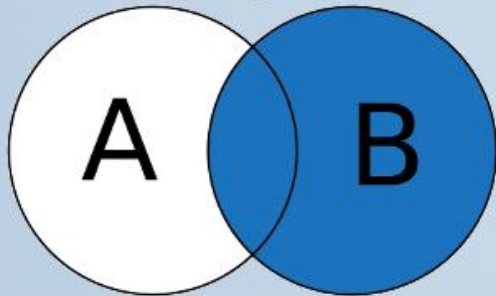
Joins

Outer Join

RIGHT JOIN o RIGHT OUTER JOIN

- Es utilizado para obtener todos los registros de la tabla B (derecha o segunda) y los registros relacionados de la tabla A

RIGHT JOIN



```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```

SQL

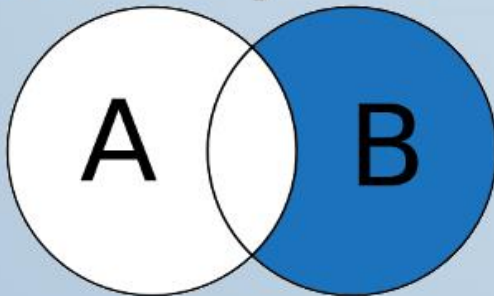
Joins

Outer Join

RIGHT JOIN o RIGHT OUTER JOIN

- Se puede condicionar para obtener los registros de la tabla B que no se relacionan con la tabla A

RIGHT JOIN



```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE A.key IS NULL
```

SQL

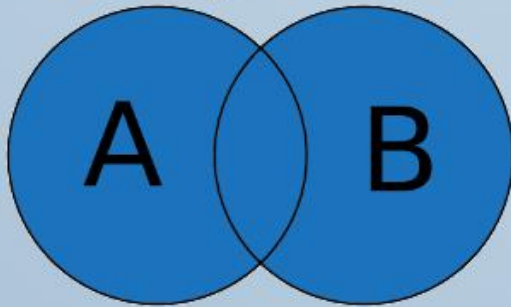
Joins

Outer Join

FULL JOIN o FULL OUTER JOIN

- Es utilizado para obtener todos los registros relacionados de ambas tablas y (UNION) los registros que no están relacionados de ambas tablas

FULL JOIN



```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key
```


SQL

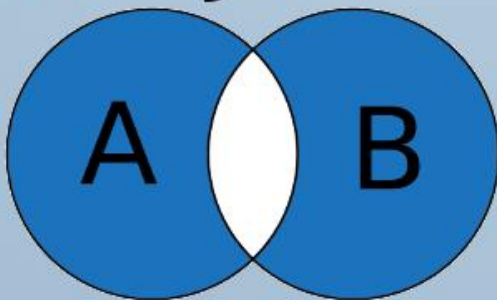
Joins

Outer Join

FULL JOIN o FULL OUTER JOIN

- Se puede condicionar para obtener los registros de ambas tablas que no se relacionan

FULL JOIN



```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```


SQL

Joins

Outer Join

FULL JOIN o FULL OUTER JOIN

- Ejemplo:

```
SELECT p.ProductID, sod.SalesOrderDetailID
FROM Production.Product p
FULL OUTER JOIN Sales.SalesOrderDetail sod
ON p.ProductID = sod.ProductID
```

	ProductID	SalesOrderDetailID
1	1	NULL
2	2	NULL
3	3	NULL
4	4	NULL
5	316	NULL
6	317	NULL

AdventureWorks2008 | 00:00:00 | 121555 rows

En este ejemplo se obtienen todas líneas de ordenes relacionadas con sus respectivos productos y (UNION) todos los productos que no fueron ordenados, más todas las líneas de órdenes que no tienen productos (en la base de datos no existen dado que no tiene sentido una línea de una orden sin un producto detallado)

FULL JOIN o FULL OUTER JOIN

- Ejemplo:

```
SELECT p.ProductID, sod.SalesOrderDetailID
FROM Production.Product p
FULL OUTER JOIN Sales.SalesOrderDetail sod
ON p.ProductID = sod.ProductID
WHERE p.ProductID IS NULL
OR sod.ProductID IS NULL
```

	ProductID	SalesOrderDetailID
1	1	NULL
2	2	NULL
3	3	NULL
4	4	NULL
5	316	NULL
6	317	NULL

AdventureWorks2008 | 00:00:00 | 238 rows

En este segundo ejemplo se obtienen todos los productos que no fueron ordenados (238), más todas las líneas de órdenes que no tienen productos (en la base de datos no existen dado que no tiene sentido una línea de una orden sin un producto detallado)

SQL

Joins

Outer Join

FULL JOIN o FULL OUTER JOIN

- Ejemplo:

```
SELECT p.ProductID, sod.SalesOrderDetailID
FROM Production.Product p
FULL OUTER JOIN Sales.SalesOrderDetail sod
ON p.ProductID = sod.ProductID
WHERE p.ProductID IS NULL
OR sod.ProductID IS NULL
UNION
SELECT ProductID, SalesOrderDetailID
FROM Sales.SalesOrderDetail
```

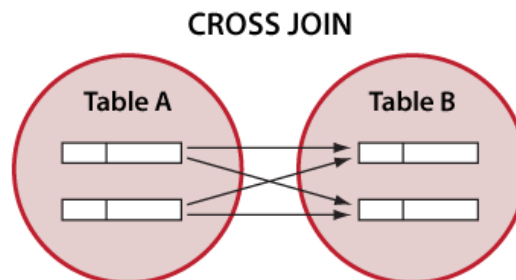
	ProductID	SalesOrderDetailID
1	794	49098
2	781	68424
3	871	56851
4	921	44806
5	865	71528
6	870	87730

AdventureWorks2008 | 00:00:00 | 121555 rows

En este último ejemplo se obtienen todos los productos que no fueron ordenados (238), más todas las líneas de órdenes que no tienen productos (en la base de datos no existen dado que no tiene sentido una línea de una orden sin un producto detallado) UNION todas las líneas de órdenes (121317) **TOTAL: 121555**

CROSS JOIN

- Una sentencia CROSS JOIN que no tiene una cláusula WHERE produce el producto cartesiano de las tablas implicadas en la unión
- El tamaño del conjunto resultado de un producto cartesiano es el número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla



SQL

Joins

Cross Join

CROSS JOIN

- Ejemplo:

```
SELECT p.BusinessEntityID, t.Name AS Territory
FROM Sales.SalesPerson p
CROSS JOIN Sales.SalesTerritory t
ORDER BY p.BusinessEntityID
```

	BusinessEntityID	Territory
1	274	Northwest
2	274	Northeast
3	274	Central
4	274	Southwest
5	274	Southeast
6	274	Canada

AdventureWorks2008 | 00:00:00 | 170 rows

En este ejemplo se obtiene el producto cartesiano entre el conjunto de Vendedores (17)y los territorios(10) **Resultado: 170**

CROSS JOIN

- Ejemplo:

```
SELECT p.BusinessEntityID, t.Name AS Territory
FROM Sales.SalesPerson p
CROSS JOIN Sales.SalesTerritory t
WHERE p.TerritoryID = t.TerritoryID
ORDER BY p.BusinessEntityID
```

	BusinessEntityID	Territory
1	275	Northeast
2	276	Southwest
3	277	Central
4	278	Canada
5	279	Southeast
6	280	Northwest

AdventureWorks2008 | 00:00:00 | 14 rows

En este ejemplo, se puede apreciar que si se utiliza la cláusula WHERE el CROSS JOIN se comporta como un INNER JOIN

Resumen Módulo V

- SQL – Joins
 - Fundamentos
 - Inner Join
 - Outer Joins
 - LEFT JOIN o LEFT OUTER JOIN
 - RIGHT JOIN o RIGHT OUTER JOIN
 - FULL JOIN o FULL OUTER JOIN
 - Cross Join