

SQL Analytics I

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

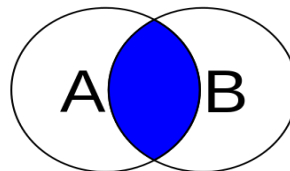
SQL

Joins

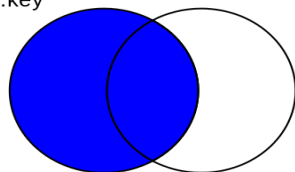
Fundamentos

- 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 reunión
 - Especificando el operador lógico de comparación a utilizar (por ejemplo, = o <>)

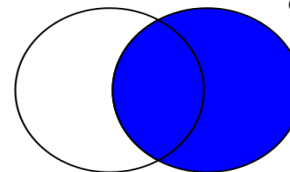
```
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key
```



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

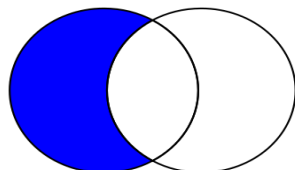


```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
```

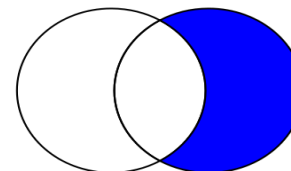


SQL JOINS

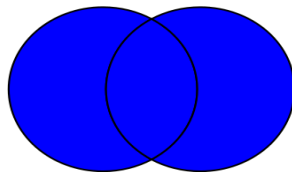
```
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL
```



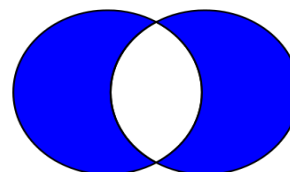
```
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
```



```
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>

Table 1 ●

| | | |
|---|--|--|
| | | |
| 1 | | |
| 2 | | |

Table 2 ●

| | | |
|---|--|--|
| | | |
| 1 | | |
| 3 | | |
| 4 | | |

Outer Join ●

| | | | | |
|---|--|--|--|--|
| | | | | |
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

Inner Join ●

| | | | | |
|---|--|--|--|--|
| | | | | |
| 1 | | | | |

Left Join ●

| | | | | |
|---|--|--|--|--|
| | | | | |
| 1 | | | | |
| 2 | | | | |

Union ●+●

| | | |
|---|--|--|
| | | |
| 1 | | |
| 2 | | |
| 1 | | |
| 3 | | |
| 4 | | |

Cross Join ●

| | | | | |
|---|--|--|---|--|
| | | | | |
| 1 | | | 1 | |
| 1 | | | 3 | |
| 1 | | | 4 | |
| 2 | | | 1 | |
| 2 | | | 3 | |
| 2 | | | 4 | |

Fuente: <https://dataschool.com/how-to-teach-people-sql/sql-join-types-explained-visually/>

SQL

Joins

Ejemplo

- Supongamos una academia donde se imparten clases, en consecuencia habrá cursos, profesores y alumnos.

TABLA CURSOS

| ID_CURSO | TITULO |
|----------|----------------------------|
| 1 | Programación PHP |
| 2 | Modelos abstracto de datos |
| 3 | SQL desde cero |
| 4 | Dibujo técnico |
| 5 | SQL avanzado |

TABLA ALUMNOS

| ID_ALUMNO | NOMBRE | APELLIDOS | F_NACIMIENTO |
|-----------|----------|----------------|--------------|
| 1 | Pablo | Hernandez Mata | 1995-03-14 |
| 2 | Jeremias | Santo Lote | 1993-07-12 |
| 3 | Teresa | Lomas Trillo | 1989-06-19 |
| 4 | Marta | Fuego García | 1992-11-23 |
| 5 | Sergio | Ot Dirmet | 1991-04-21 |
| 6 | Carmen | Dilma Perna | 1987-12-04 |

TABLA PROFESORES

| ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
|----------|----------|---------------|--------------|
| 1 | Federico | Gasco Daza | 1975-04-23 |
| 2 | Ana | Saura Trenzo | 1969-08-02 |
| 3 | Rosa | Honrosa Pérez | 1980-09-05 |

SQL

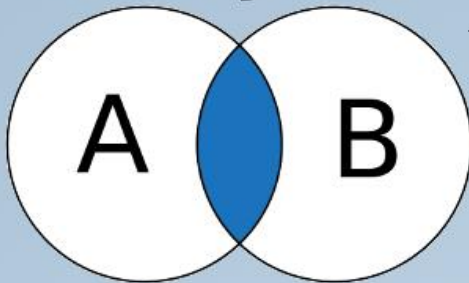
Joins

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

- Consulta que realiza la reunión entre los profesores y los cursos que imparte cada uno usando INNER JOIN / ON:
- `select * from CURSOS C inner join PROFESORES P on C.ID_PROFE = P.ID_PROFE`

| ID_CURSO | TITULO | ID_PROFE | TABLA PROFESORES | | | |
|----------|----------------------------|----------|------------------|----------|---------------|--------------|
| 1 | Programación PHP | 3 | ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
| 2 | Modelos abstracto de datos | 3 | 1 | Federico | Gasco Daza | 1975-04-23 |
| 3 | SQL desde cero | 1 | 2 | Ana | Saura Trenzo | 1969-08-02 |
| 4 | Dibujo técnico | 2 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 5 | SQL avanzado | | | | | |

| ID_CURSO | TITULO | ID_PROFE | ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
|----------|----------------------------|----------|----------|----------|---------------|--------------|
| 1 | Programación PHP | 3 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 2 | Modelos abstracto de datos | 3 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 3 | SQL desde cero | 1 | 1 | Federico | Gasco Daza | 1975-04-23 |
| 4 | Dibujo técnico | 2 | 2 | Ana | Saura Trenzo | 1969-08-02 |

SQL

Joins

Inner Join

INNER JOIN o JOIN

- Ejemplo:

```
SELECT soh.SalesOrderID, soh.OrderDate,  
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 | OrderDate | OrderQty |
|----|--------------|-------------------------|----------|
| 1 | 47395 | 2002-09-01 00:00:00.000 | 7 |
| 2 | 53472 | 2003-09-01 00:00:00.000 | 7 |
| 3 | 47027 | 2002-08-01 00:00:00.000 | 6 |
| 4 | 46660 | 2002-07-01 00:00:00.000 | 6 |
| 5 | 46671 | 2002-07-01 00:00:00.000 | 5 |
| 6 | 47369 | 2002-09-01 00:00:00.000 | 5 |
| 7 | 51123 | 2003-07-01 00:00:00.000 | 5 |
| 8 | 53530 | 2003-09-01 00:00:00.000 | 5 |
| 9 | 57141 | 2003-11-01 00:00:00.000 | 5 |
| 10 | 57157 | 2003-11-01 00:00:00.000 | 4 |

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.

SQL

Joins

Inner Join

INNER JOIN o JOIN

- Ejemplo:

```
SELECT DISTINCT p.ProductID, p.Name, p.ListPrice, sd.UnitPrice AS  
'Precio de Venta'  
FROM Sales.SalesOrderDetail AS sd  
JOIN Production.Product AS p  
ON sd.ProductID = p.ProductID  
WHERE p.ProductID = 718
```

| | Name | ListPrice | Precio de Venta |
|---|-------------------------|-----------|-----------------|
| 1 | HL Road Frame - Red, 44 | 1431.50 | 758.0759 |
| 2 | HL Road Frame - Red, 44 | 1431.50 | 780.8182 |
| 3 | HL Road Frame - Red, 44 | 1431.50 | 858.90 |

En este ejemplo se obtiene el precio de lista del producto con ID 718 y el precio al que fue vendido en su momento

SQL

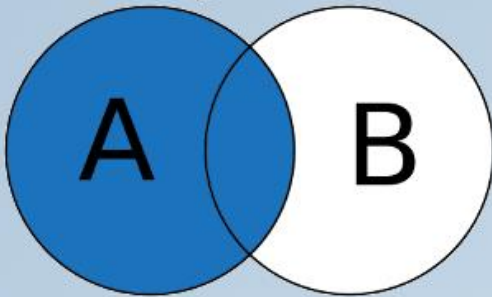
Joins

left 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

left Join

- Consulta que muestra los cursos y sus profesores aunque el curso no tenga profesor asignado:
- `select *from CURSOS C left outer join
PROFESORES P on C.ID_PROFE = P.ID_PROFE`

| ID_CURSO | TITULO | ID_PROFE | TABLA PROFESORES | | | |
|----------|----------------------------|----------|------------------|----------|---------------|--------------|
| 1 | Programación PHP | 3 | | | | |
| 2 | Modelos abstracto de datos | 3 | | | | |
| 3 | SQL desde cero | 1 | | | | |
| 4 | Dibujo técnico | 2 | | | | |
| 5 | SQL avanzado | | | | | |
| ID_CURSO | TITULO | ID_PROFE | ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
| 3 | SQL desde cero | 1 | 1 | Federico | Gasco Daza | 1975-04-23 |
| 4 | Dibujo técnico | 2 | 2 | Ana | Saura Trenzo | 1969-08-02 |
| 1 | Programación PHP | 3 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 2 | Modelos abstracto de datos | 3 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 5 | SQL avanzado | | | | | |

SQL

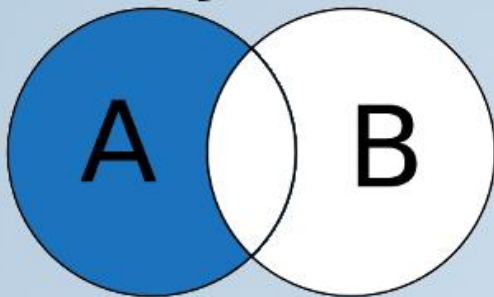
Joins

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

left Join

- Consulta que muestra los cursos que no tienen profesor asignado:
- `select *from CURSOS C left outer join
PROFESORES P on C.ID_PROFE = P.ID_PROFE
where P.ID_PROF is null`

| ID_CURSO | TITULO | ID_PROFE | TABLA PROFESORES | | | |
|----------|----------------------------|----------|------------------|----------|---------------|--------------|
| 1 | Programación PHP | 3 | ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
| 2 | Modelos abstracto de datos | 3 | 1 | Federico | Gasco Daza | 1975-04-23 |
| 3 | SQL desde cero | 1 | 2 | Ana | Saura Trenzo | 1969-08-02 |
| 4 | Dibujo técnico | 2 | 3 | Rosa | Honrosa Pérez | 1980-09-05 |
| 5 | SQL avanzado | | | | | |
| ID_CURSO | TITULO | ID_PROFE | ID_PROFE | NOMBRE | APELLIDOS | F_NACIMIENTO |
| 5 | SQL avanzado | | | | | |

SQL

Joins

left Join

LEFT JOIN o LEFT OUTER JOIN

- Ejemplo:

```
SELECT *
FROM Production.Product p
LEFT JOIN Production.ProductInventory i
ON p.ProductID = i.ProductID
```

| | ProductID | Name | ProductNumber | MakeFlag | FinishedGoodsFlag | Color | SafetyStockLevel |
|-----|-----------|-----------------------|---------------|----------|-------------------|--------|------------------|
| 951 | 915 | ML Touring Seat/S... | SE-T762 | 0 | 1 | NULL | 500 |
| 952 | 916 | HL Touring Seat/Sa... | SE-T924 | 0 | 1 | NULL | 500 |
| 953 | 916 | HL Touring Seat/Sa... | SE-T924 | 0 | 1 | NULL | 500 |
| 954 | 916 | HL Touring Seat/Sa... | SE-T924 | 0 | 1 | NULL | 500 |
| 955 | 917 | LL Mountain Frame ... | FR-M21S-42 | 1 | 1 | Silver | 500 |
| 956 | 918 | LL Mountain Frame ... | FR-M21S-44 | 1 | 1 | Silver | 500 |

1141 rows

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

SQL

Joins

left Join

LEFT JOIN o LEFT OUTER JOIN

- Ejemplo:

```
SELECT *  
FROM Production.Product p  
LEFT JOIN Production.ProductInventory i  
on p.ProductID = i.ProductID  
WHERE i.ProductID is null
```

| | ProductID | Name | ProductNumber | MakeFlag | FinishedGoodsFlag | Color | SafetyStockLevel |
|---|-----------|---------------------------|---------------|----------|-------------------|-------|------------------|
| 1 | 680 | HL Road Frame - Black, 58 | FR-R92B-58 | 1 | 1 | Black | 500 |
| 2 | 706 | HL Road Frame - Red, 58 | FR-R92R-58 | 1 | 1 | Red | 500 |
| 3 | 717 | HL Road Frame - Red, 62 | FR-R92R-62 | 1 | 1 | Red | 500 |
| 4 | 718 | HL Road Frame - Red, 44 | FR-R92R-44 | 1 | 1 | Red | 500 |
| 5 | 719 | HL Road Frame - Red, 48 | FR-R92R-48 | 1 | 1 | Red | 500 |

72 rows

En este ejemplo se obtienen todos los productos que no tienen inventario (72)

SQL

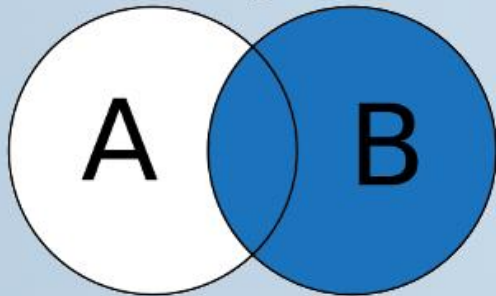
Joins

right 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

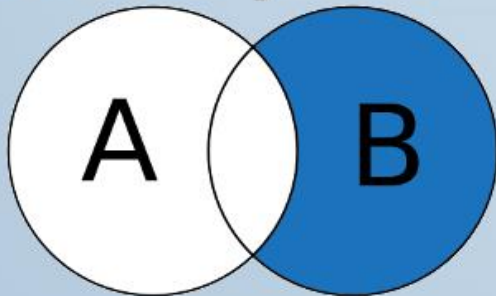
Joins

right 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

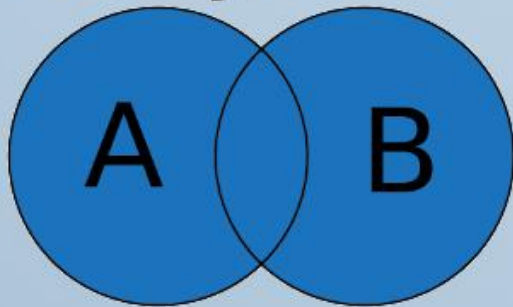
Joins

full 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

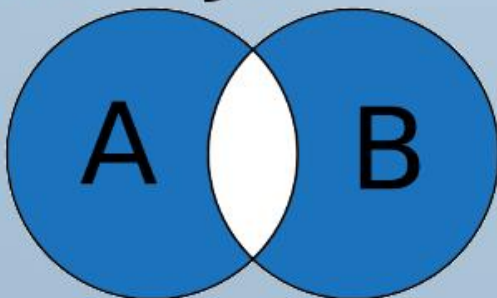
Joins

full 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

full 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)

SQL

Joins

full 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
```

| | 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**

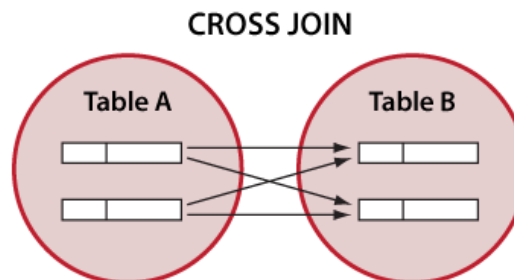
SQL

Joins

Cross Join

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



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