

Material Imprimible

Curso de SQL Server

## Módulo 7: Conectando distintos conjuntos de datos II

### JOIN / INNER JOIN

La sentencia INNER JOIN es la sentencia JOIN por defecto, y consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición.

**SELECT \* FROM tabla1 INNER JOIN tabla2 WHERE tabla1.columna1 = tabla2.columna1**

#### Ejemplo SQL INNER JOIN

Tabla personas, con la clave primaria "per "

| per | nombre  | apellido1 | apellido2 | dep |
|-----|---------|-----------|-----------|-----|
| 1   | ANTONIO | PEREZ     | GOMEZ     | 1   |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ | 2   |
| 3   | PEDRO   | RUIZ      | GONZALEZ  | 2   |

Tabla "departamentos", con la clave primaria "dep"

| dep | departamento   |
|-----|----------------|
| 1   | ADMINISTRACION |
| 2   | INFORMATICA    |
| 3   | COMERCIAL      |

**SELECT nombre, apellido1, departamento FROM personas INNER JOIN departamentos WHERE personas.dep = departamentos.dep**

| nombre  | apellido1 | departamento   |
|---------|-----------|----------------|
| ANTONIO | PEREZ     | ADMINISTRACION |
| ANTONIO | GARCIA    | INFORMATICA    |
| PEDRO   | RUIZ      | INFORMATICA    |

JOIN puede ser usada en SQL Management Studio pero INNER JOIN es compatible con la mayoría de los sistemas que usan SQL.

### LEFT JOIN

La sentencia **LEFT JOIN** combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de la primera tabla, incluso aunque no cumplan la condición.

```
SELECT * FROM tabla1
LEFT JOIN tabla2
WHERE tabla1.columna1 = tabla2.columna1
```

Ejemplo de SQL LEFT JOIN

Tabla personas, con la clave primaria "per "

| per | nombre  | apellido1 | apellido2 | dep |
|-----|---------|-----------|-----------|-----|
| 1   | ANTONIO | PEREZ     | GOMEZ     | 1   |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ | 2   |
| 3   | PEDRO   | RUIZ      | GONZALEZ  | 4   |

Tabla "departamentos", con la clave primaria "dep"

| dep | departamento   |
|-----|----------------|
| 1   | ADMINISTRACION |
| 2   | INFORMATICA    |
| 3   | COMERCIAL      |

```
SELECT nombre, apellido1, departamento
FROM personas
LEFT JOIN departamentos
WHERE personas.dep = departamentos.dep
```

| nombre  | apellido1 | departamento   |
|---------|-----------|----------------|
| ANTONIO | PEREZ     | ADMINISTRACION |
| ANTONIO | GARCIA    | INFORMATICA    |
| PEDRO   | RUIZ      |                |

Aunque el departamento '4' de PEDRO RUIZ no existe en la tabla de departamentos, devolverá la fila con esa columna 'departamento' en blanco.

### RIGHT JOIN

La sentencia RIGHT JOIN combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de la segunda tabla, incluso aunque no cumplan la condición.

En algunas bases de datos, la sentencia RIGHT JOIN es igual a RIGHT OUTER JOIN

```
SELECT * FROM tabla1 RIGHT JOIN tabla2 WHERE tabla1.columna1 = tabla2.columna1
```

### EJEMPLO SQL RIGHT JOIN

| per | nombre  | apellido1 | apellido2 | dep |
|-----|---------|-----------|-----------|-----|
| 1   | ANTONIO | PEREZ     | GOMEZ     | 1   |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ | 2   |
| 3   | PEDRO   | RUIZ      | GONZALEZ  | 4   |

Tabla "departamentos", con la clave primaria "dep"

| dep | departamento   |
|-----|----------------|
| 1   | ADMINISTRACION |
| 2   | INFORMATICA    |
| 3   | COMERCIAL      |

```
SELECT nombre, apellido1, departamento
FROM personas
RIGHT JOIN departamentos
WHERE personas.dep = departamentos.dep
```

| nombre  | apellido1 | departamento   |
|---------|-----------|----------------|
| ANTONIO | PEREZ     | ADMINISTRACION |
| ANTONIO | GARCIA    | INFORMATICA    |
|         |           | COMERCIAL      |

Aunque no exista ninguna persona del departamento 'COMERCIAL' (3), esta fila aparecerá con las otras columnas en blanco.

### FULL JOIN

La sentencia **FULL JOIN** combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de las dos tablas, aunque no cumplan la condición.

La sentencia **FULL JOIN** es la unión de **LEFT JOIN** y **RIGHT JOIN**

```
SELECT * FROM tabla1 FULL JOIN tabla2 WHERE tabla1.columna1 = tabla2.columna1
```

**EJEMPLO SQL FULL JOIN**

| per | nombre  | apellido1 | apellido2 | dep |
|-----|---------|-----------|-----------|-----|
| 1   | ANTONIO | PEREZ     | GOMEZ     | 1   |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ | 2   |
| 3   | PEDRO   | RUIZ      | GONZALEZ  | 4   |

Tabla "departamentos", con la clave primaria "dep"

| dep | departamento   |
|-----|----------------|
| 1   | ADMINISTRACION |
| 2   | INFORMATICA    |
| 3   | COMERCIAL      |

```

SELECT nombre, apellido1, departamento
FROM personas
FULL JOIN departamentos
WHERE personas.dep = departamentos.dep

```

| nombre  | apellido1 | departamento   |
|---------|-----------|----------------|
| ANTONIO | PEREZ     | ADMINISTRACION |
| ANTONIO | GARCIA    | INFORMATICA    |
| PEDRO   | RUIZ      | COMERCIAL      |

### ALTER TABLE

La sentencia SQL ALTER se utiliza para añadir, eliminar o modificar columnas de una tabla.

#### Sintaxis SQL ALTER

Para añadir una nueva columna a una tabla:

```
ALTER TABLE nombretabla
```

```
ADD nombrecolumna tipodatocolumna
```

Para borrar una columna de una tabla:

**ALTER TABLE nombretabla**

**DROP COLUMN nombrecolumna**

Para modificar el tipo de dato de una columna de una tabla:

**ALTER TABLE nombretabla**

**ALTER COLUMN nombrecolumna tipodatocolumna**

**Ejemplos de SQL ALTER**

| per | nombre  | apellido1 | apellido2 |
|-----|---------|-----------|-----------|
| 1   | ANTONIO | PEREZ     | GOMEZ     |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ |
| 3   | PEDRO   | RUIZ      | GONZALEZ  |

Dada la siguiente tabla de 'personas', queremos añadir una nueva columna, denominada 'fechadenacimiento'

**ALTER TABLE personas**

**ADD fechadenacimiento date**

| per | nombre  | apellido1 | apellido2 | fechadenacimiento |
|-----|---------|-----------|-----------|-------------------|
| 1   | ANTONIO | PEREZ     | GOMEZ     |                   |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ |                   |
| 3   | PEDRO   | RUIZ      | GONZALEZ  |                   |

Si queremos modificar el tipo de dato de la columna 'fecha', y ponerle tipo 'year' en lugar de tipo 'date'

**ALTER TABLE personas**

**ALTER COLUMN fechadenacimiento year**

Si queremos borrar la columna 'fechadenacimiento', y dejarlo igual que al principio

**ALTER TABLE personas**

**DROP COLUMN fechadenacimiento**

| per | nombre  | apellido1 | apellido2 |
|-----|---------|-----------|-----------|
| 1   | ANTONIO | PEREZ     | GOMEZ     |
| 2   | ANTONIO | GARCIA    | RODRIGUEZ |
| 3   | PEDRO   | RUIZ      | GONZALEZ  |

Fuente: <https://javieroto.wordpress.com/tag/group-by/page/3/>