

## 5- Combinación de tablas

### Objetivos:

- Utiliza sentencias para unir los datos de diferentes tablas.

### Recursos:

- Microsoft SQL Server Management Studio
- Guías prácticas.
- Script de bases de datos.

### Introducción

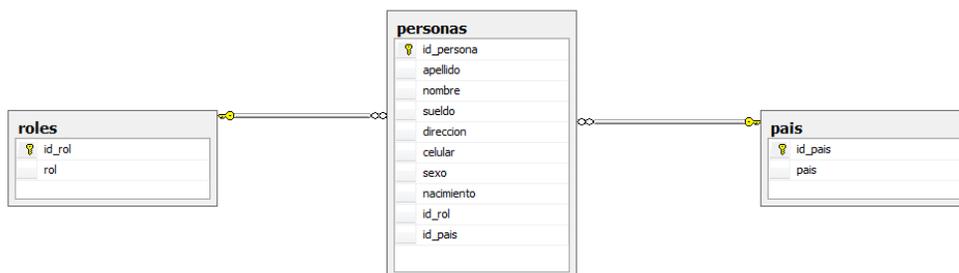
Una vez se tienen creada la base de datos y hemos creado las relaciones, ahora debemos aprender a realizar consultas de dos o más tablas para obtener resultados solicitados, los cuales pueden tener condiciones WHERE o HAVING, y otros operadores como AVG, SUM, GROUP BY y los operadores AND y OR.

### JOIN

La sentencia JOIN en SQL permite combinar registros de dos o más tablas en una base de datos relacional. En el Lenguaje de Consultas Estructurado (SQL) hay tres tipos de JOIN: interno, externo y cruzado.

En casos especiales una tabla puede unirse a sí misma, produciendo una auto-combinación, SELF-JOIN. Matemáticamente, JOIN es composición relacional, la operación fundamental en el álgebra relacional, y, generalizando, es una función de composición.

Usaremos las siguientes tres tablas para probar las diferentes opciones, al final de esta guía encontraras el script para crear las tablas con sus registros.



## INNER JOIN (interna)

Con esta operación se calcula el producto cruzado de todos los registros; así cada registro en la tabla A es combinado con cada registro de la tabla B; pero sólo permanecen aquellos registros en la tabla combinada que satisfacen las condiciones que se especifiquen. Este es el tipo de JOIN más utilizado, por lo que es considerado el tipo de combinación predeterminado.

Es necesario tener especial cuidado cuando se combinan columnas con valores nulos NULL, ya que el valor nulo no se combina con otro valor o con otro nulo, excepto cuando se le agregan predicados tales como IS NULL o IS NOT NULL.

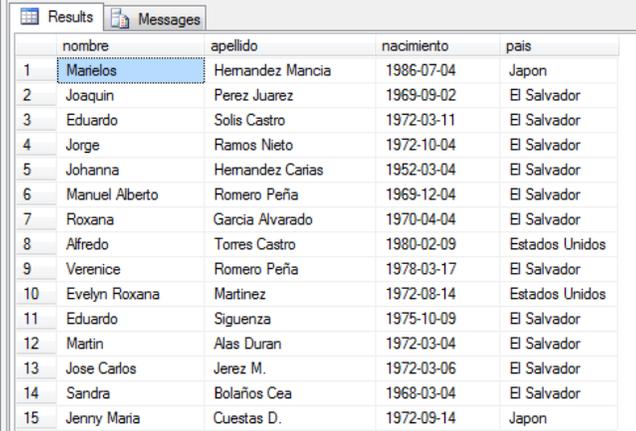
Vamos a realizar el primer ejemplo combinando las tablas “personas” y “pais”, y mostraremos los campos de la tabla personas: nombre, apellido y nacimiento, y de la tabla país: país.

Si ejecutamos el siguiente comando:

```
USE SCRUM  
GO
```

```
SELECT personas.nombre, personas.apellido, personas.nacimiento, pais.pais  
FROM personas INNER JOIN pais ON personas.id_pais = pais.id_pais
```

Observe que los ocupamos el nombre las tablas unido al de los campos, para identificar cada uno de ellos, además se usa la palabra clave “ON” la cual sirve para unir las tablas pro su campo de relación.



	nombre	apellido	nacimiento	pais
1	Marielos	Hernandez Mancía	1986-07-04	Japon
2	Joaquin	Perez Juarez	1969-09-02	El Salvador
3	Eduardo	Solis Castro	1972-03-11	El Salvador
4	Jorge	Ramos Nieto	1972-10-04	El Salvador
5	Johanna	Hernandez Carías	1952-03-04	El Salvador
6	Manuel Alberto	Romero Peña	1969-12-04	El Salvador
7	Roxana	García Alvarado	1970-04-04	El Salvador
8	Alfredo	Torres Castro	1980-02-09	Estados Unidos
9	Verenice	Romero Peña	1978-03-17	El Salvador
10	Evelyn Roxana	Martínez	1972-08-14	Estados Unidos
11	Eduardo	Siguenza	1975-10-09	El Salvador
12	Martin	Alas Duran	1972-03-04	El Salvador
13	Jose Carlos	Jerez M.	1972-03-06	El Salvador
14	Sandra	Bolaños Cea	1968-03-04	El Salvador
15	Jenny Maria	Cuevas D.	1972-09-14	Japon

Podemos apreciar que en el resultado aparece el nombre del país y no su código, además solo presentan los datos que cumplen exactamente con la condición de unión.

Otro ejemplo es para unir tres tablas a la vez, en este caso aumentaremos una INNER JOIN más a la consulta y una condición ON también.

```
USE SCRUM
GO
```

```
SELECT personas.nombre, personas.apellido, personas.nacimiento, roles.rol, pais.pais
FROM personas INNER JOIN pais ON personas.id_pais = pais.id_pais
INNER JOIN roles ON personas.id_rol = roles.id_rol
```



	nombre	apellido	nacimiento	rol	pais
1	Marielos	Hernandez Mancia	1986-07-04	Team	Japon
2	Joaquin	Perez Juarez	1969-09-02	Team	El Salvador
3	Eduardo	Solis Castro	1972-03-11	Team	El Salvador
4	Jorge	Ramos Nieto	1972-10-04	Steakholders	El Salvador
5	Johanna	Hernandez Carias	1952-03-04	ScrumMaster	El Salvador
6	Manuel Alberto	Romero Peña	1969-12-04	ProductOwner	El Salvador
7	Roxana	Garcia Alvarado	1970-04-04	ScrumMaster	El Salvador
8	Alfredo	Torres Castro	1980-02-09	Team	Estados Unidos
9	Verenice	Romero Peña	1978-03-17	Team	El Salvador
10	Evelyn Roxana	Martinez	1972-08-14	Team	Estados Unidos
11	Eduardo	Siguenza	1975-10-09	Team	El Salvador
12	Martin	Alas Duran	1972-03-04	Team	El Salvador
13	Jose Carlos	Jerez M.	1972-03-06	Steakholders	El Salvador
14	Sandra	Bolaños Cea	1968-03-04	Team	El Salvador
15	Jenny Maria	Cuestas D.	1972-09-14	Team	Japon

El resultado es la unión de las tres tablas, la opción INNER JOIN es la más usada, sin embargo hay otras opciones.

### LEFT JOIN o LEFT OUTER JOIN. (externa)

El conjunto de resultados de una combinación externa izquierda incluye todas las filas de la tabla de la izquierda especificada en la cláusula LEFT OUTER y no sólo aquellas en las que coincidan las columnas combinadas. Cuando una fila de la tabla de la izquierda no tiene filas coincidentes en la tabla de la derecha, la fila asociada del conjunto de resultados contiene valores NULL en todas las columnas de la lista de selección que procedan de la tabla de la derecha.

Si ejecutamos la siguiente consulta con la base de datos ejemplo:

```
USE SCRUM
GO
```

```
SELECT personas.nombre, personas.apellido, personas.sueldo, pais.pais
FROM pais
LEFT JOIN personas
ON personas.id_pais = pais.id_pais
```

	nombre	apellido	sueldo	pais
1	Joaquin	Perez Juarez	850.00	El Salvador
2	Eduardo	Solis Castro	600.00	El Salvador
3	Jorge	Ramos Nieto	550.00	El Salvador
4	Johanna	Hernandez Carias	1800.00	El Salvador
5	Manuel Alberto	Romero Peña	0.00	El Salvador
6	Roxana	Garcia Alvarado	1600.00	El Salvador
7	Verenice	Romero Peña	400.00	El Salvador
8	Eduardo	Siguenza	1120.00	El Salvador
9	Martin	Alas Duran	1200.00	El Salvador
10	Jose Carlos	Jerez M.	589.00	El Salvador
11	Sandra	Bolaños Cea	244.00	El Salvador
12	NULL	NULL	NULL	Guatemala
13	NULL	NULL	NULL	Costa Rica
14	Marielos	Hernandez Mancia	958.00	Japon
15	Jenny Maria	Cuestas D.	1089.00	Japon
16	Alfredo	Tomés Castro	450.00	Estados Unidos
17	Evelyn Roxana	Martinez	340.00	Estados Unidos
18	NULL	NULL	NULL	Argentina
19	NULL	NULL	NULL	Mexico
20	NULL	NULL	NULL	Uruguay
21	NULL	NULL	NULL	Canada
22	NULL	NULL	NULL	Chile

Al usar esta opción nos permite unir todos los datos de la tabla de la izquierda al LEFT JOIN con todos los datos que coincidan o no con los de la derecha, por ese motivo los países que no tiene correlación en la otra tabla aparecen con nulos.

En este caso hay 7 países que no tiene datos registrados en la tabla “personas”.

Es muy importante tener en cuenta la posición de las tablas, ya que si cambiamos el orden, se cambiaria el resultado.

### RIGHT JOIN o RIGHT OUTER JOIN. (externa)

Una combinación externa derecha es lo contrario de una combinación externa izquierda. Se devuelven todas las filas de la tabla de la derecha. Cada vez que una fila de la tabla de la derecha no tenga correspondencia en la tabla de la izquierda, se devuelven valores NULL para la tabla de la izquierda.

Para probar esta opción utilizaremos la siguiente consulta.

```
USE SCRUM
GO
```

```
SELECT personas.nombre, personas.apellido, personas.sueldo, roles.rol
FROM personas
RIGHT JOIN roles
ON personas.id_rol = roles.id_rol
```

El resultado mostrara que se a combinado la tabla derecha con la izquierda y como la tabla “roles” tiene un registro que no corresponde con la tabla “personas”, ha colocado un valor nulo.

	nombre	apellido	sueldo	rol
1	Johanna	Hernandez Carias	1800.00	ScrumMaster
2	Roxana	Garcia Alvarado	1600.00	ScrumMaster
3	Manuel Alberto	Romero Peña	0.00	ProductOwner
4	Manielos	Hernandez Mancia	958.00	Team
5	Joaquin	Perez Juarez	850.00	Team
6	Eduardo	Solis Castro	600.00	Team
7	Alfredo	Torres Castro	450.00	Team
8	Verenice	Romero Peña	400.00	Team
9	Evelyn Roxana	Martinez	340.00	Team
10	Eduardo	Siguenza	1120.00	Team
11	Martin	Alas Duran	1200.00	Team
12	Sandra	Bolaños Cea	244.00	Team
13	Jenny Maria	Cuestas D.	1089.00	Team
14	Jorge	Ramos Nieto	550.00	Steakholders
15	Jose Carlos	Jerez M.	589.00	Steakholders
16	NULL	NULL	NULL	Tester

## FULL JOIN o FULL OUTER JOIN. (externa)

Una combinación externa completa devuelve todas las filas de las tablas de la izquierda y la derecha. Cada vez que una fila no tenga coincidencia en la otra tabla, las columnas de la lista de selección de la otra tabla contendrán valores NULL. Cuando haya una coincidencia entre las tablas, la fila completa del conjunto de resultados contendrá los valores de datos de las tablas base.

Probemos esta opción con la siguiente consulta.

```
USE SCRUM
GO
```

```
SELECT personas.nombre, personas.apellido, personas.sueldo, pais.pais
FROM pais
FULL JOIN personas
ON personas.id_pais = pais.id_pais
```

	nombre	apellido	sueldo	pais
1	Joaquin	Perez Juarez	850.00	El Salvador
2	Eduardo	Solis Castro	600.00	El Salvador
3	Jorge	Ramos Nieto	550.00	El Salvador
4	Johanna	Hernandez Carias	1800.00	El Salvador
5	Manuel Alberto	Romero Peña	0.00	El Salvador
6	Roxana	Garcia Alvarado	1600.00	El Salvador
7	Verenice	Romero Peña	400.00	El Salvador
8	Eduardo	Siguenza	1120.00	El Salvador
9	Martin	Alas Duran	1200.00	El Salvador
10	Jose Carlos	Jerez M.	589.00	El Salvador
11	Sandra	Bolaños Cea	244.00	El Salvador
12	NULL	NULL	NULL	Guatemala
13	NULL	NULL	NULL	Costa Rica
14	Manielos	Hernandez Mancia	958.00	Japon
15	Jenny Maria	Cuestas D.	1089.00	Japon
16	Alfredo	Torres Castro	450.00	Estados Unidos
17	Evelyn Roxana	Martinez	340.00	Estados Unidos
18	NULL	NULL	NULL	Argentina
19	NULL	NULL	NULL	Mexico
20	NULL	NULL	NULL	Uruguay
21	NULL	NULL	NULL	Canada
22	NULL	NULL	NULL	Chile

La consulta devuelve todo el registro de ambas tablas, y lo que no corresponde pondrá NULL.

## CROSS JOIN. (cruzado)

Una combinación cruzada que no tenga una cláusula WHERE genera el producto cartesiano de las tablas involucradas en la combinación. El tamaño del conjunto de resultados de un producto cartesiano es igual al número de filas de la primera tabla multiplicado por el número de filas de la segunda tabla.

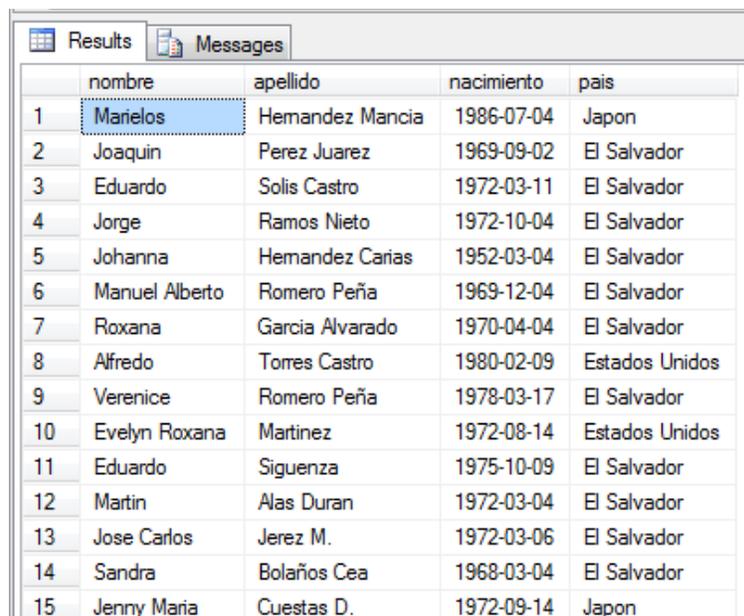
Cross Join no necesita que se le incorpore la cláusula "ON", En el siguiente ejemplo genera una combinación cruzada de las tablas personas y país.

```
USE SCRUM
GO
```

```
SELECT personas.nombre, personas.apellido, personas.nacimiento, pais.pais
FROM personas CROSS JOIN pais
```

Sin embargo esta consulta nos dará 150 resultados que se obtiene de multiplicar los registros de una tabla con otra. Para mejorar esta consultado le incorporaremos un WHERE y la consulta tendrá como resultado el equivalente a un INNER JOIN.

```
SELECT personas.nombre, personas.apellido, personas.nacimiento, pais.pais
FROM personas CROSS JOIN pais WHERE personas.id_pais = pais.id_pais
```



	nombre	apellido	nacimiento	pais
1	Marielos	Hernandez Mancia	1986-07-04	Japon
2	Joaquin	Perez Juarez	1969-09-02	El Salvador
3	Eduardo	Solis Castro	1972-03-11	El Salvador
4	Jorge	Ramos Nieto	1972-10-04	El Salvador
5	Johanna	Hernandez Carias	1952-03-04	El Salvador
6	Manuel Alberto	Romero Peña	1969-12-04	El Salvador
7	Roxana	Garcia Alvarado	1970-04-04	El Salvador
8	Alfredo	Torres Castro	1980-02-09	Estados Unidos
9	Verenice	Romero Peña	1978-03-17	El Salvador
10	Evelyn Roxana	Martinez	1972-08-14	Estados Unidos
11	Eduardo	Siguenza	1975-10-09	El Salvador
12	Martin	Alas Duran	1972-03-04	El Salvador
13	Jose Carlos	Jerez M.	1972-03-06	El Salvador
14	Sandra	Bolaños Cea	1968-03-04	El Salvador
15	Jenny Maria	Cuestas D.	1972-09-14	Japon

Ahora es importante el probar consultadas incorporando mas condiciones, para lo cual realice las siguientes consultas y verifique su resultado y comportamientos para cada caso.

## Ejercicios:

- Pruebe las opciones de unión incorporando las condiciones vistas en las dos guías anteriores, como WHERE,
- Realice pruebas con la base de datos NORTWIND.
- Realice los ejercicios que le asignará el instructor

## Datos ejemplo:

```
CREATE DATABASE SCRUM
GO

USE SCRUM
GO

CREATE TABLE roles
(
  id_rol int IDENTITY(1,1),
  rol varchar(20) not null,
  CONSTRAINT PK_roles PRIMARY KEY (id_rol)
)

INSERT INTO roles VALUES ('ScrumMaster')
INSERT INTO roles VALUES ('ProductOwner')
INSERT INTO roles VALUES ('Team')
INSERT INTO roles VALUES ('Steakholders')
INSERT INTO roles VALUES ('Tester')

CREATE TABLE pais
(
  id_pais int IDENTITY(1,1),
  pais varchar(25),
  CONSTRAINT PK_pais PRIMARY KEY (id_pais)
)

INSERT INTO pais VALUES ('El Salvador')
INSERT INTO pais VALUES ('Guatemala')
INSERT INTO pais VALUES ('Costa Rica')
INSERT INTO pais VALUES ('Japon')
INSERT INTO pais VALUES ('Estados Unidos')
INSERT INTO pais VALUES ('Argentina')
INSERT INTO pais VALUES ('Mexico')
INSERT INTO pais VALUES ('Uruguay')
INSERT INTO pais VALUES ('Canada')
INSERT INTO pais VALUES ('Chile')

CREATE TABLE personas
(
  id_persona int IDENTITY(1,1),
  apellido varchar(40) not null,
  nombre varchar(40) not null,
  sueldo smallmoney,
  direccion varchar(100),
  celular varchar(8),
  sexo varchar(1),
  nacimiento date,
  id_rol int not null,
  id_pais int not null,
  CONSTRAINT PK_persona PRIMARY KEY (id_persona),
  CONSTRAINT FK_rol FOREIGN KEY (id_rol)
  REFERENCES roles(id_rol),
  CONSTRAINT FK2_pais FOREIGN KEY (id_pais)
  REFERENCES pais(id_pais)
)

INSERT INTO personas VALUES ('Hernandez Mancia', 'Marielos', 958, '', '67567555', 'F', '1986/7/4', 3, 4)
```

```
INSERT INTO personas VALUES ('Perez Juarez', 'Joaquin', 850, '', '34567564', 'M', '1969/9/2', 3, 1)
INSERT INTO personas VALUES ('Solis Castro', 'Eduardo', 600, '', '12327523', 'M', '1972/3/11', 3, 1)
INSERT INTO personas VALUES ('Ramos Nieto', 'Jorge', 550, '', '34567174', 'M', '1972/10/4', 4, 1)
INSERT INTO personas VALUES ('Hernandez Carias', 'Johanna', 1800, '', '34234809', 'F', '1952/3/4', 1, 1)
INSERT INTO personas VALUES ('Romero Peña', 'Manuel Alberto', 0, '', '34567342', 'M', '1969/12/4', 2, 1)
INSERT INTO personas VALUES ('Garcia Alvarado', 'Roxana', 1600, '', '34345564', 'F', '1970/4/4', 1, 1)
INSERT INTO personas VALUES ('Torres Castro', 'Alfredo', 450, '', '34565674', 'M', '1980/2/9', 3, 5)
INSERT INTO personas VALUES ('Romero Peña', 'Verenice', 400, '', '34567589', 'F', '1978/3/17', 3, 1)
INSERT INTO personas VALUES ('Martinez', 'Evelyn Roxana', 340, '', '34556764', 'F', '1972/8/14', 3, 5)
INSERT INTO personas VALUES ('Siguenza', 'Eduardo', 1120, '', '21467564', 'M', '1975/10/9', 3, 1)
INSERT INTO personas VALUES ('Alas Duran', 'Martin', 1200, '', '34562364', 'M', '1972/3/4', 3, 1)
INSERT INTO personas VALUES ('Jerez M.', 'Jose Carlos', 589, '', '34563352', 'M', '1972/3/6', 4, 1)
INSERT INTO personas VALUES ('Bolaños Cea', 'Sandra', 244, '', '67557564', 'F', '1968/3/4', 3, 1)
INSERT INTO personas VALUES ('Cuestas D.', 'Jenny Maria', 1089, '', '34567564', 'F', '1972/9/14', 3, 4)
```