

OPERACIONES ENTRE TABLAS

Ing. Duver Rene Acosta



UNIVERSITARIA AGUSTINIANA
UNIAGUSTINIANA

DATOS DE MÚLTIPLES TABLAS



PROPÓSITO



Comprender la creación de sentencias SQL y las relaciones de múltiples tablas

BASE DE DATOS XYZ



NEGOCIO

Almacén XYZ

OBJETO SOCIAL

Comercializa
productos de aseo

BASE DE DATOS XYZ

cliente	
idCliente	VARCHAR(20)
Nombre	VARCHAR(50)
Apellido	VARCHAR(50)
Fecha_nacimiento	DATE
Direccion	VARCHAR(45)
Telefono	VARCHAR(15)
ciudad_Codigo	VARCHAR(15)

ciudad	
Codigo	VARCHAR(15)
Nombre	VARCHAR(50)
Superficie	FLOAT
Poblacion	INT(11)
Altura	INT(11)

Crear la DB, hacer la relación y llenar las tablas con tres registros.

SQL JOIN

La instrucción SQL JOIN se utiliza para combinar dos o más tablas, tomando un campo común de las dos de una base de datos relacional.



Se desea conocer el nombre, el apellido, el teléfono y la ciudad de los clientes.

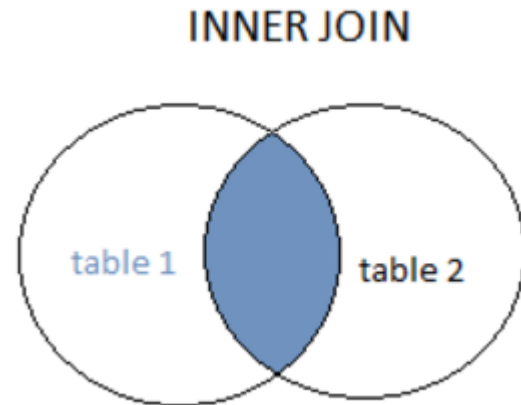
```
SELECT cliente.idCliente, cliente.nombre,  
cliente.telefono, ciudad.nombre FROM  
cliente, ciudad WHERE  
cliente.ciudad_codigo=ciudad.codigo;
```

TABLE 1, TABLE N

SQL INNER JOIN

(Join simple). Un SQL INNER JOIN devuelve todos los registros de varias tablas que cumplen con la condición.

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name=table2.column_name;
```



INNER JOIN

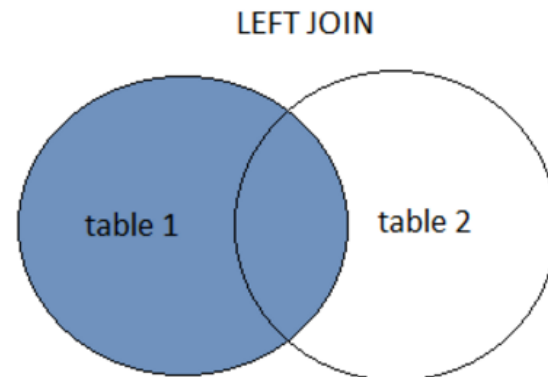
Se desea conocer el nombre, el apellido, el teléfono y la ciudad de los clientes.

```
SELECT cliente.idCliente, cliente.nombre,  
cliente.telefono, ciudad.nombre FROM cliente  
INNER JOIN ciudad WHERE  
cliente.ciudad=ciudad.codigo;
```

SQL LEFT JOIN

LEFT JOIN devuelve todas los registros de la tabla de la izquierda (table1), con las correspondientes de la tabla de la derecha (table2). El resultado es NULL en la parte de la derecha cuando no hay registros que correspondan con la condición.

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name=table2.column_name;
```



LEFT JOIN

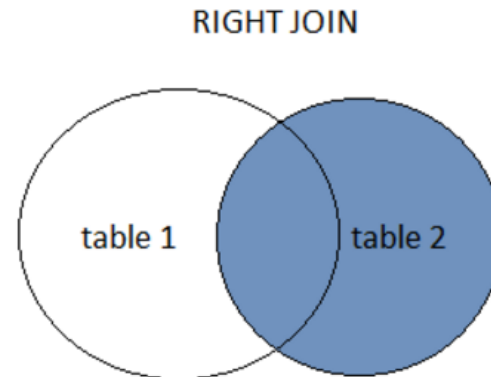
Se desea conocer el nombre, el apellido, el teléfono y la ciudad de los clientes.

```
SELECT cliente.idCliente, cliente.nombre,  
cliente.telefono, ciudad.nombre FROM cliente  
LEFT JOIN ciudad ON  
cliente.ciudad=ciudad.codigo;
```

RIGHT JOIN

La instrucción RIGHT JOIN devuelve todos los registros de la tabla de la derecha (table2), y todos los registros correspondientes de la tabla de la izquierda (table1). El resultado será NULL cuando no haya registros correspondientes de la tabla de la izquierda.

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name=table2.column_name;
```



RIGHT JOIN

Se desea conocer el nombre, el apellido, el teléfono y la ciudad de los clientes.

```
SELECT cliente.idCliente, cliente.nombre,  
cliente.telefono, ciudad.nombre FROM cliente  
RIGHT JOIN ciudad ON  
cliente.ciudad=ciudad.codigo;
```

OTROS JOIN

CROSS JOIN: Se utiliza en SQL-SERVER

para realizar consultas de unión. Combina cada fila de una tabla con cada fila de otra tabla

FULL OUTER JOIN: Combina los resultados de dos o más tablas, tengan o no coincidencia entre sí.

OTROS JOIN

EQUI JOIN: Cuando se usan operadores, tales como $< o >$ no se puede clasificar en este rango.

SELF JOIN: Cuando se usa un join sobre la tabla que lo utiliza.

CONCATENAR

Une el contenido de dos o más campos o expresiones.

```
SELECT CONCAT(Nombre , ' ', Apellido) FROM  
cliente WHERE Nombre='Carolina';
```

```
SELECT CONCAT('My', 'S', 'QL');
```


SUBSTRING

Se utiliza para tomar una parte de los datos almacenados.

```
SELECT SUBSTR(Nombre,5) FROM cliente  
WHERE Nombre='Maria Camila';
```