

Módulo 2

Código: 484

Bases de datos

Técnico Superior en Desarrollo de
Aplicaciones Multiplataforma



UNIDAD 6

DQL

Guía didáctica profesor



Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Obtener datos de múltiples tablas*

- Es habitual necesitar en una consulta, datos que se encuentran distribuidos en varias tablas.
- Las bases de datos relacionales se basan en que los datos se distribuyen en tablas que se pueden relacionar mediante un campo.
- Ese campo es el que permite integrar y unir los datos de las tablas

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Producto Cartesiano*

Un producto cartesiano o cruzado, combina todos los registros de una tabla con todos los registros de la otra tabla.

```
SELECT * FROM municipio, provincia;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Producto Cartesiano*

Un producto cartesiano o cruzado, combina todos los registros de una tabla con todos los registros de la otra tabla.

SELECT * FROM municipio, provincia;

	codigo [PK] character varying (5)	nombre character varying (40)	provincia character (2)
1	CSTLL	Castellón de la Plana	CA
2	VLLRL	Villarreal	CA
3	BRRN	Burriana	CA
4	VLLDX	Vall de Uxó	CA
5	VNRZ	Vinaroz	CA
6	BNCRL	Benicarló	CA
7	VLNC	Valencia	VA
8	TRRNT	Torrente	VA
9	GND	Gandía	VA
10	PTRN	Paterna	VA
11	SGNT	Sagunto	VA
12	LCNT	Alicante	AL
13	LCH	Elche	AL
14	TRRVJ	Torreveja	AL
15	RHL	Orihuela	AL
16	BNDRM	Benidorm	AL

X

	codigo [PK] character (2)	nombre character varying (30)
1	VA	Valencia
2	AL	Alicante
3	CA	Castellón

Sesiones 10 a 14. Contenidos teóricos

	codigo character varying (5)	nombre character varying (40)	provincia character (2)	codigo character (2)	nombre character varying (30)
1	CSTLL	Castellón de la Plana	CA	VA	Valencia
2	CSTLL	Castellón de la Plana	CA	AL	Alicante
3	CSTLL	Castellón de la Plana	CA	CA	Castellón
4	VLLRL	Villarreal	CA	VA	Valencia
5	VLLRL	Villarreal	CA	AL	Alicante
6	VLLRL	Villarreal	CA	CA	Castellón
7	BRRN	Burriana	CA	VA	Valencia
8	BRRN	Burriana	CA	AL	Alicante
9	BRRN	Burriana	CA	CA	Castellón
10	VLLDX	Vall de Uxó	CA	VA	Valencia
11	VLLDX	Vall de Uxó	CA	AL	Alicante
12	VLLDX	Vall de Uxó	CA	CA	Castellón
13	VNRZ	Vinaroz	CA	VA	Valencia
14	VNRZ	Vinaroz	CA	AL	Alicante
15	VNRZ	Vinaroz	CA	CA	Castellón
16	BNCRL	Benicarló	CA	VA	Valencia
17	BNCRL	Benicarló	CA	AL	Alicante
18	BNCRL	Benicarló	CA	CA	Castellón
19	VLNC	Valencia	VA	VA	Valencia
20	VLNC	Valencia	VA	AL	Alicante
21	VLNC	Valencia	VA	CA	Castellón

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – Obtener datos de múltiples tablas

De todas las filas anteriores, hay algunas que resultan interesantes y otras que no. Por ejemplo, Castellón de la Plana no se encuentra en la provincia de Valencia o Alicante. Las filas que interesan son las que están relacionadas.

¿Cómo seleccionarlas? con una condición de relación en la cláusula where. De esta forma la sentencia siguiente:

```
SELECT * FROM municipio, provincia  
WHERE provincia = provincia.codigo ;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – Obtener datos de múltiples tablas

	codigo character varying (5)	nombre character varying (40)	provincia character (2)	codigo character (2)	nombre character varying (30)
1	CSTLL	Castellón de la Plana	CA	CA	Castellón
2	VLLRL	Villarreal	CA	CA	Castellón
3	BRRN	Burriana	CA	CA	Castellón
4	VLLDX	Vall de Uxó	CA	CA	Castellón
5	VNRZ	Vinaroz	CA	CA	Castellón
6	BNCRL	Benicarló	CA	CA	Castellón
7	VLNC	Valencia	VA	VA	Valencia
8	TRRNT	Torrente	VA	VA	Valencia
9	GND	Gandía	VA	VA	Valencia
10	PTRN	Paterna	VA	VA	Valencia
11	SGNT	Sagunto	VA	VA	Valencia
12	LCNT	Alicante	AL	AL	Alicante
13	LCH	Elche	AL	AL	Alicante
14	TRRVJ	Torrevieja	AL	AL	Alicante
15	RHL	Orihuela	AL	AL	Alicante
16	BNDRM	Benidorm	AL	AL	Alicante

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Obtener datos de múltiples tablas*

```
SELECT * FROM municipio, provincia  
WHERE provincia = provincia.codigo ;
```

Si en el WHERE existen dos columnas con el mismo nombre que puedan producir algún tipo de ambigüedad, es necesario indicar delante a qué tabla estamos haciendo referencia.

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Alias de Columna*

Los nombres de las tablas aparecen varias veces en la sentencia SQL. Si, además, los nombres son largos, la escritura de las sentencias pasa a convertirse en algo largo y aburrido. Para evitarlo, SQL proporciona un mecanismo para construir fácilmente alias de tablas

```
SELECT m.nombre, pr.nombre  
FROM municipio m, provincia pr  
WHERE m.provincia = pr.codigo;
```

Podemos seguir utilizando ALIAS en el SELECT:

```
SELECT m.nombre AS municipio, pr.nombre AS provincia  
FROM municipio m, provincia pr  
WHERE m.provincia = pr.codigo;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

SINTAXIS ANTERIOR A SQL 1999

WHERE une las tablas por el campo clave primaria y foránea

```
SELECT m.nombre municipio, p.nombre provincia  
FROM municipio m, provincia p  
WHERE m.provincia = p.codigo;
```

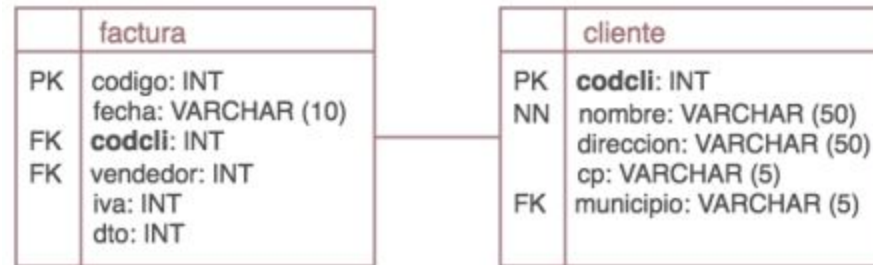
SINTAXIS SQL 1999

- NATURAL JOIN
- INNER JOIN ON
- CROSS JOIN

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Natural JOIN*

Establece una relación de igualdad entre las tablas a través de los campos que tengan el mismo nombre en ambas tablas. Supongamos el siguiente modelo Relacional:



En el ejemplo anterior, sería posible escribir una sentencia que muestre el código de factura, la fecha y el nombre del cliente destinatario de la factura utilizando NATURAL JOIN.

```
SELECT f.codigo, f.fecha, c.nombre FROM factura f NATURAL JOIN cliente c ;
```

Es equivalente a:

```
SELECT f.codigo, f.fecha, c.nombre FROM factura f, cliente c WHERE f.codcli = c.codcli;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Inner JOIN on*

Establece relaciones cuya **condición se establece manualmente**, lo que permite realizar asociaciones más complejas o bien asociaciones cuyos campos en las tablas no tienen el mismo nombre.

Ejemplo. *Escribir una sentencia que muestre el código de ticket, la fecha y el nombre del cliente destinatario del mismo.*

```
SELECT t.codigo, t.fecha, c.nombre  
      FROM ticket t  
INNER JOIN cliente c ON t.cliente = c.codigo;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS – *Cross JOIN*

Realiza un **producto cruzado** entre las tablas indicadas. Eso significa que cada tupla de la primera tabla se combina con cada tupla de la segunda tabla. No es una operación muy utilizada, aunque posibilita resolver consultas extremadamente complicadas.

Ejemplo. *Escribir una sentencia que obtenga el producto cartesiano de la tabla clientes y vendedores.*

```
SELECT * FROM cliente CROSS JOIN vendedor;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Ejemplo. *Escribir una sentencia que muestre el nombre de cada municipio y el de su provincia.*

```
SELECT * FROM municipio m  
INNER JOIN provincia p ON m.provincia = p.codigo
```

Ejemplo. *Escribir una sentencia que muestre el código de ticket, la fecha y el nombre del vendedor.*

```
SELECT t.codigo, t.fecha, v.nombre FROM ticket t  
INNER JOIN vendedor v ON t.vendedor = v.codigo;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Ejemplo. *Escribir una sentencia que muestre el código de ticket, el número de línea, el código del artículo, la descripción del artículo y el número de unidades vendidas en dicha línea para todas las líneas del ticket cuyo código es 7.*

```
SELECT lt.ticket, lt.nlinea, a.codigo, a.descripcion, lt.cant  
FROM linea_ticket lt  
INNER JOIN articulo a ON lt.articulo = a.codigo  
WHERE ticket = 7;
```


Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Ejemplo. *Escribir una sentencia que muestre el nombre del vendedor y el nombre del municipio en que reside para aquellos vendedores cuyo código se encuentra entre 1 y 3 inclusive.*

```
SELECT v.nombre, m.nombre  
FROM vendedor v  
INNER JOIN municipio m ON v.municipio = m.codigo  
WHERE v.codigo BETWEEN 1 AND 3;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Se concatenan dos tablas entre sí y, después, se concatena el resultado con la tercera tabla.

Ejemplo. *Escribir una sentencia que muestre el código y fecha de cada ticket junto al nombre del cliente y el nombre del vendedor.*

```
SELECT t.codigo, c.nombre AS cliente, v.nombre AS vendedor  
      FROM ticket t  
      INNER JOIN vendedor v ON t.vendedor = v.codigo  
      INNER JOIN cliente c ON t.cliente = c.codigo;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Ejemplo. *Escribir una sentencia que muestre el código y fecha de cada ticket, junto al nombre y dirección completa (calle, código postal, municipio y provincia) del cliente.*

```
SELECT t.codigo, t.fecha, c.nombre, c.direccion, c.cp, m.nombre, p.nombre  
FROM ticket t  
INNER JOIN cliente c ON t.cliente = c.codigo  
INNER JOIN municipio m ON c.municipio = m.codigo  
INNER JOIN provincia p ON m.provincia = p.codigo;
```

Sesiones 10 a 14. Contenidos teóricos

CONCATENACIÓN DE TABLAS

Ejemplo. *Escribir una sentencia que muestre el código y nombre de cada vendedor junto al total de tickets que ha vendido.*

```
SELECT v.codigo, v.nombre, COUNT(v.codigo)  
FROM vendedor v  
INNER JOIN ticket t ON t.vendedor = v.codigo  
GROUP BY v.codigo;
```

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS

Es una variante de la concatenación interna que permite que no se pierdan filas de una tabla, aunque no exista ninguna fila relacionada en la otra tabla.

Ejemplo. *Supongamos que se desea mostrar el nombre de cada cliente junto el municipio en el que reside. Tenemos las siguientes tablas:*

TABLA CLIENTES		
CODCLI	NOMBRE	CODMUN
101	Alberto	1000
102	Carlos	1000
103	Juan	
104	Pedro	1001

TABLA MUNICIPIOS	
CODMUN	NOMBRE
1000	CELLA
1001	OLOCAU
1002	JUMILLA

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS

Al hacer la unión interna de las tablas obtenemos la siguiente tabla resultado.

```
SELECT c.nombre cliente, p.nombre municipio FROM clientes c  
INNER JOIN municipios m ON m.codmun=c.codmun
```

CLIENTE	MUNICIPIO
ALBERTO	CELLA
CARLOS	CELLA
PEDRO	OLOCAU

En el resultado final no aparecen todos los datos. No aparece:

- El cliente Juan, cuyo código de municipio es nulo, no aparece, porque al realizar la concatenación de tablas no existe ningún municipio cuyo código sea nulo.
- El municipio de Jumilla no aparece porque no existe ningún cliente cuyo código de municipio sea el de Jumilla (código 1002).

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS

En ocasiones resulta muy interesante que no se pierda ninguna fila de una u otra tabla al realizar la concatenación. Para evitarlo es necesario hacer una concatenación externa. Esta es muy similar a la interna sólo que evita que se pierdan filas que no están relacionadas.

Ejemplos en los que es necesario realizar una concatenación externa:

- Listado de clientes con sus direcciones completas. Si un cliente no tiene código de municipio y se realiza una concatenación interna, éste no aparece en el listado, lo cual puede dar lugar a creer que no existe en la base de datos.
- Listado de facturas con sus importes y nombres de clientes. Si una factura no tiene código de cliente y se realiza una concatenación interna, ésta no aparece en el listado.
- Listado de artículos con las unidades vendidas durante el año pasado. Si un artículo no ha sido vendido, no aparecerá. En muchas ocasiones puede resultar interesante que sí aparezca dicho artículo, pero teniendo como venta 0 unidades.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS

- Tabla1 LEFT OUTER JOIN Tabla2
- Tabla1 RIGHT OUTER JOIN Tabla2
- Tabla1 FULL OUTER JOIN Tabla2

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – LEFT OUTER JOIN

- El resultado contiene todas las filas de la Tabla1.
- Las filas de la Tabla1 que se relacionan con alguna de las filas de la Tabla2 aparecen concatenadas en el resultado.
- Las filas de la Tabla1 que no se relacionan con ninguna fila de la Tabla2 aparecen en el resultado concatenadas con una fila de nulos.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – LEFT OUTER JOIN

Ejemplo. Mostrar el código de ticket, fecha y nombre de los clientes sin que se pierda ningún ticket, aunque su cliente sea null.

```
SELECT t.codigo, fecha, c.nombre FROM ticket t LEFT JOIN cliente c ON  
t.cliente = c.codigo;
```

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – RIGHT OUTER JOIN

- El resultado contiene todas las filas de la Tabla2.
- Las filas de la Tabla2 que se relacionan con alguna de las filas de la Tabla1 aparecen concatenadas en el resultado.
- Las filas de la Tabla2 que no se relacionan con ninguna fila de la Tabla1 aparecen en el resultado concatenadas con una fila de nulos.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – RIGHT OUTER JOIN

Ejemplo. Mostrar el nombre y dirección de los clientes junto con el nombre de su municipio sin perder aquellos municipios que no tienen clientes.

```
SELECT c.nombre, direccion, m.nombre FROM cliente c RIGHT JOIN municipio  
m ON c.municipio = m.codigo;
```

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – FULL OUTER JOIN

- El resultado contiene todas las filas de las tablas Tabla1 y Tabla2.
- Realiza las dos operaciones anteriores simultáneamente.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS – FULL OUTER JOIN

Si se desea que no se pierda ningún cliente ni tampoco ningún municipio del resultado final, hay que recurrir a la concatenación externa completa.

Ejemplo. Mostrar el código de ticket, la fecha y el nombre del cliente incluso cuando el ticket no tiene cliente incluyendo todos los clientes aunque no aparezcan en ningún ticket.

```
SELECT t.codigo, t.fecha, c.nombre FROM ticket t FULL JOIN cliente c ON t.cliente = c.codigo;
```

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN DE TABLAS - EQUIVALENCIAS

- A LEFT OUTER JOIN B es completamente equivalente a B RIGHT OUTER JOIN A.
- A RIGHT OUTER JOIN B es equivalente a B LEFT OUTER JOIN A.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TABLAS - EQUIVALENCIAS

Ejemplo: Escribir una consulta que devuelva el código de ticket, línea, código y descripción del artículo para todas las líneas del ticket 108, aunque el código de artículo fuese nulo.

```
SELECT l.ticket, l.nlinea, a.codigo, a.descripcion  
FROM linea_ticket l  
LEFT JOIN articulo a ON a.codigo=l.articulo  
WHERE l.ticket = '108';
```

```
SELECT l.ticket, l.nlinea, a.codigo, a.descripcion  
FROM articulo a  
RIGHT JOIN linea_ticket l ON a.codigo=l.articulo  
WHERE l.ticket = '108';
```


Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN DE TABLAS Y AGRUPACIÓN

Se necesita mostrar el nombre de cada municipio y el número de clientes residiendo en cada uno de ellos. Esta consulta necesita concatenación de dos tablas y una agrupación.

Solución con concatenación interna.

```
SELECT m.nombre, COUNT(*) FROM cliente AS c INNER JOIN municipio AS m ON c.municipio  
= m.codigo GROUP BY m.nombre;
```

Los municipios que no tienen ningún cliente no
aparecerán

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN DE TABLAS Y AGRUPACIÓN

Si queremos que aparezcan todos los municipios hay que utilizar concatenación externa.
Solución con concatenación externa.

```
SELECT m.nombre, COUNT(*) FROM cliente AS c RIGHT JOIN municipio AS m ON c.municipio =  
m.codigo GROUP BY m.nombre;
```

- **El resultado no es correcto.** Muestra que Alcoy tiene un cliente.
- Lo que ocurre es que el municipio de Alcoy es concatenado con un cliente a null y la función de grupo count(*) cuenta la fila y devuelve 1.

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN DE TABLAS Y AGRUPACIÓN

La solución es sustituir el operador count(*) que devuelve 1 en el caso de Alcoy por el operador count(c.codigo) que devolverá cero en dicho caso pues el código del cliente para dicho municipio es nulo.
Solución CORRECTA con concatenación externa.

```
SELECT m.nombre, COUNT(c.codigo) FROM cliente AS c RIGHT JOIN municipio AS m ON c.municipio  
= m.codigo GROUP BY m.nombre;
```

Sesiones 15 a 19. Contenidos teóricos

CONCATENACIÓN EXTERNA DE TRES O MÁS TABLAS

Se realiza de manera similar a las concatenación de dos tablas.

Ejemplo: Escribir una consulta que devuelva el código y fecha de cada ticket junto al nombre del vendedor que realizó la venta y el nombre del cliente destinatario del ticket. No se debe perder ningún ticket aunque no tenga código de vendedor o código de cliente.

```
SELECT t.codigo, t.fecha, v.nombre, c.nombre  
FROM ticket t  
LEFT JOIN cliente c ON t.cliente = c.codigo  
LEFT JOIN vendedor v ON t.vendedor = v.codigo;
```