

Consultas anidadas

Tema 3

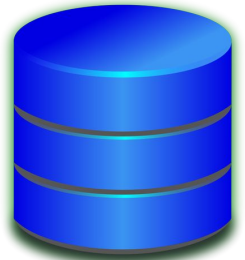
Bases de Datos - Grado en Ingeniería Informática
Antonio Balderas Alberico

Comentarios

- Transparencias correspondientes al **capítulo 4** del documento “Apuntes de prácticas” del Campus Virtual:
 - <https://av03-21-22.uca.es/moodle/mod/resource/view.php?id=72860>
- Para el lenguaje MySQL, se aportan referencias web oficiales donde se amplía y especifica el uso de las diferentes instrucciones.

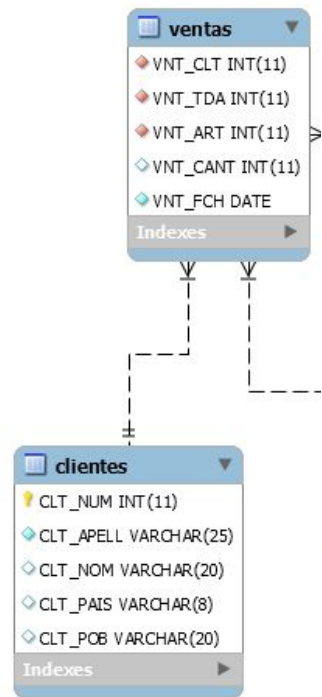
Introducción

- Definimos los criterios para filtrar en las cláusulas WHERE o HAVING
- ¿Podemos usar como criterio de selección el resultado de otra consulta?



Introducción

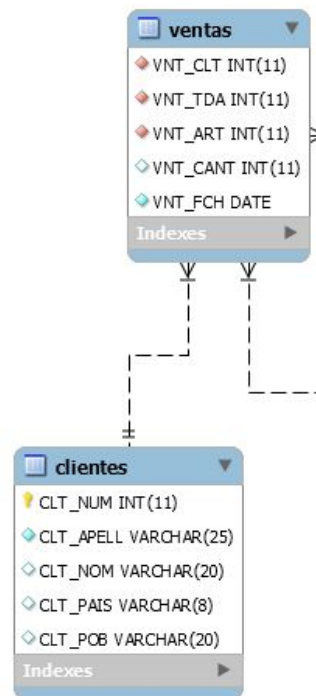
- ¿Qué clientes compraron el 2 de noviembre de 2019?



Introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```



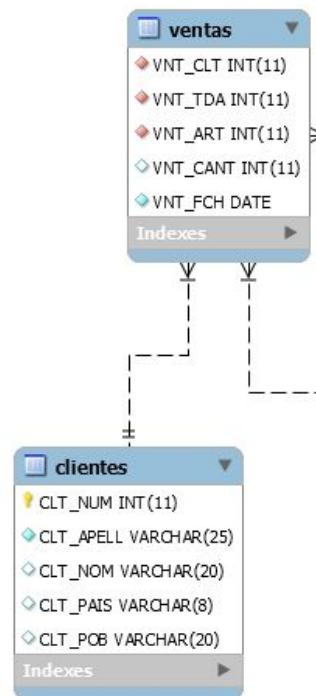
Introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```

Resultado → clientes 6 y 13

- ¿Quiénes son esos clientes?



Introducción

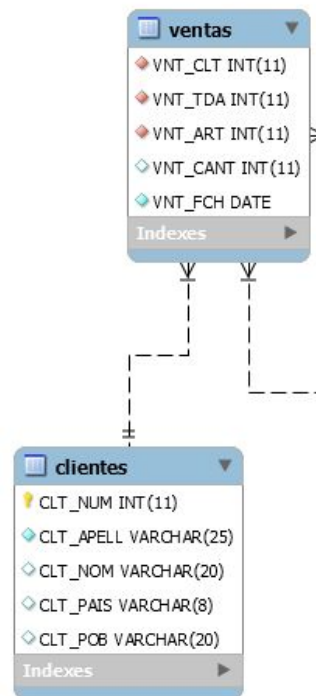
- ¿Qué clientes compraron el 2 de noviembre de 2019?

```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```

Resultado → clientes 6 y 13

- ¿Quiénes son esos clientes?

```
SELECT * FROM clientes WHERE clt_num IN (6,13);
```



Introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

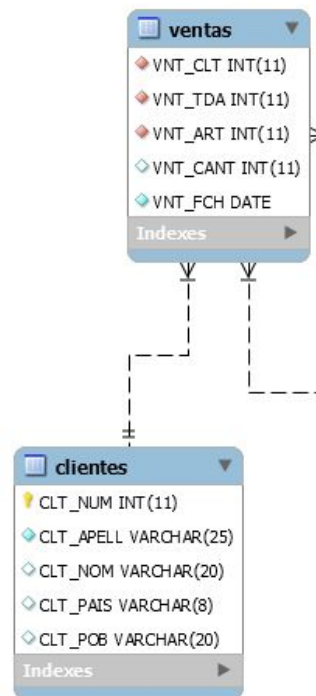
```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```

Resultado → clientes 6 y 13

- ¿Quiénes son esos clientes?

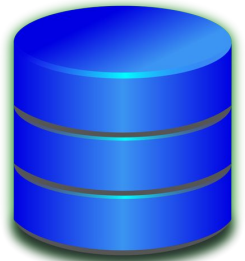
```
SELECT * FROM clientes WHERE clt_num IN (6,13);
```

6	Souris	...
13	Cortes	...



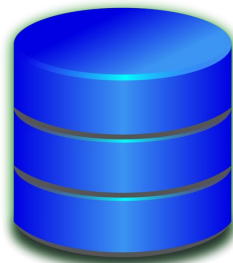
Resumen de consultas anidadas

- Se pueden anidar consultas tanto en WHERE como en HAVING
- Sin límite de anidación
- Sólo se muestran datos de la consulta de nivel superior
- Devolución un valor o múltiples valores o filas (in, all, any)
- Comparación múltiples columnas
- Consultas correlacionadas



Sintaxis

```
SELECT columnas/expresiones/funciones  
FROM tabla  
WHERE columna/expresión/función  
operador_de_comparación  
    (SELECT columna/expresión/función  
     FROM tabla  
     [WHERE ...]  
     [GROUP BY ...]  
     [HAVING ...] ) ;
```



Ejemplo de introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

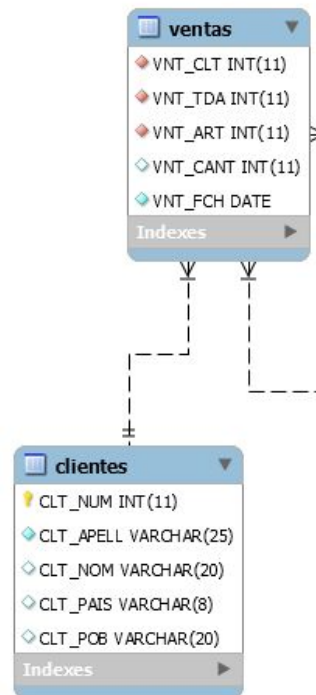
```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```

Resultado → clientes 6 y 13

- ¿Quiénes son esos clientes?

```
SELECT * FROM clientes WHERE clt_num IN (6,13);
```

6	Souris	...
13	Cortes	...



Ejemplo de introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

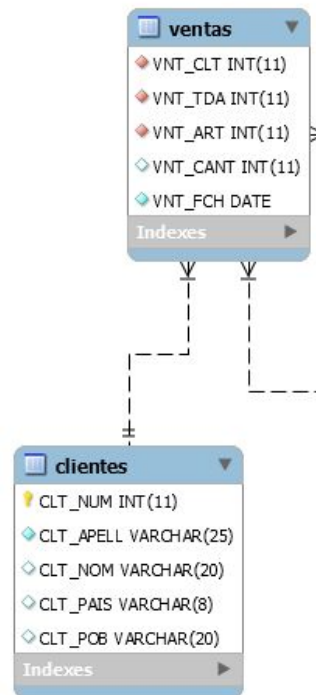
```
SELECT vnt_clt FROM ventas WHERE vnt_fch = '2019-11-02';
```

Resultado → clientes 6 y 13

- ¿Quiénes son esos clientes?

```
SELECT * FROM clientes WHERE clt_num IN (6,13);
```

6	Souris	...
13	Cortes	...

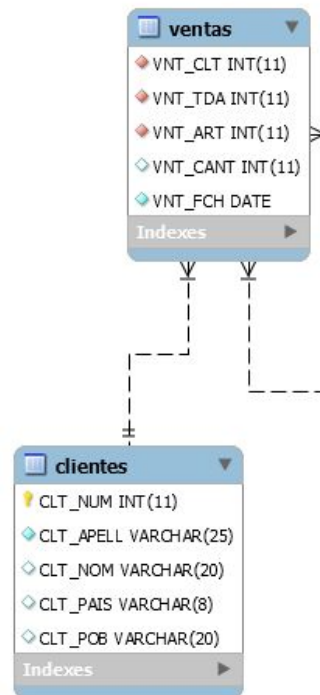


Ejemplo de introducción

- ¿Qué clientes compraron el 2 de noviembre de 2019?

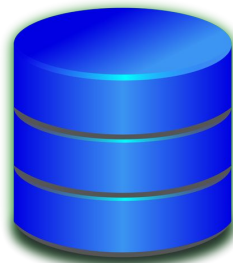
```
SELECT *  
FROM clientes  
WHERE clt_num IN  
      (SELECT vnt_clt  
      FROM ventas  
      WHERE vnt_fch = '2019-11-02');
```

6	Souris	...
13	Cortes	...



Devolución de un valor o múltiples valores

```
SELECT columnas/expresiones/funciones  
FROM tabla  
WHERE columna/expresión/función  
operador_de_comparación  
(SELECT columna/expresión/función  
  FROM tabla  
  [WHERE ...]  
  [GROUP BY ...]  
  [HAVING ...] ) ;
```



Devolución de un valor o múltiples valores

```
SELECT columnas/expresiones/funciones
FROM tabla
WHERE columna/expresión/función operador_de_comparación
(SELECT columna/expresión/función
FROM tabla
[WHERE ...]
[GROUP BY ...]
[HAVING ...] );
```

Al diseñar la consulta, debemos plantear cuántos valores puede devolver la subconsulta:

- Un valor
- Múltiples valores

Y esto es importante, porque según sea un caso u otro habrá que utilizar un operador u otro



Devolución de un valor

- La subconsulta devuelve un valor que sirve como comparación de la consulta de la cláusula WHERE principal
- La subconsulta devolverá siempre un único valor cuando:
 - Se filtra sobre la clave primaria o un campo único
 - Proyecta una función de grupo que implica a toda la tabla
- Los criterios de selección se podrán combinar mediante operadores lógicos con otros criterios basados en la comparación de columnas/funciones/expresiones o otras subconsultas.

Devolución de un valor

- Listado de artículo que son del mismo color que el artículo número 15.

```
SELECT *  
FROM articulos  
WHERE art_col =  
      (SELECT art_col  
       FROM articulos  
       WHERE art_num = 15);
```

art_num es clave primaria → la subconsulta sólo devolverá un color

Devolución de múltiples filas

- La subconsulta puede devolver múltiples filas, luego debe incluirse en la cláusula WHERE con operadores de lista (IN) o utilizando los operadores ALL o ANY
- La subconsulta puede devolver múltiples filas cuando:
 - No se filtra sobre la clave primaria o un campo único
 - No se proyecta una función de grupo que implica a toda la tabla
- Los criterios de selección se podrán combinar mediante operadores lógicos con otros criterios basados en la comparación de columnas/funciones/expresiones o otras subconsultas.

Devolución de múltiples filas

- Listado de proveedores que proporcionan artículos de color verde.

```
SELECT *  
FROM proveedores  
WHERE prv_num IN  
    (SELECT art_prv  
     FROM articulos  
     WHERE art_col = 'verde');
```

art_col no es clave primaria → la subconsulta puede devolver más de un valor

Devolución de múltiples filas

- La lista obtenida por una subconsulta puede tratarse mediante los operadores ALL (todos) o ANY (al menos uno).
- La subconsulta debe colocarse en una cláusula WHERE seguida de un operador comparativo (=, <>, !=, <, >, <=, =>) seguido por uno de los operadores ALL o ANY

Devolución de múltiples filas

- Listado de artículos cuyo precio de venta es mayor que cualquier precio de venta de los artículos de color verde

```
SELECT *  
FROM articulos  
WHERE art_pv > ALL  
    (SELECT art_pv  
     FROM articulos  
     WHERE art_col = 'verde');
```

art_col no es clave primaria → la subconsulta puede devolver más de un valor

Devolución de múltiples filas

- Listado de artículos cuyo precio de venta es mayor que algún precio de venta de los artículos de color verde

```
SELECT *  
FROM articulos  
WHERE art_pv > ANY  
      (SELECT art_pv  
       FROM articulos  
       WHERE art_col = 'verde');
```

art_col no es clave primaria → la subconsulta puede devolver más de un valor

Devolución de múltiples columnas

- Puede darse el caso de que las subconsultas devuelvan más de una columna
- En este caso, el orden de las columnas expresadas en la cláusula WHERE de la consulta de nivel superior debe coincidir con el orden de las columnas seleccionadas por la consulta de nivel inferior

Devolución de múltiples columnas

- Listado de artículos que tengan el mismo color y peso que el artículo número 10

```
SELECT *  
FROM articulos  
WHERE (art_col, art_peso) =  
      (SELECT art_col, art_peso  
       FROM articulos  
       WHERE art_num = 10);
```

art_num es clave primaria → la subconsulta devolverá un único valor

Subconsultas correlacionadas

- Una consulta anidada se evalúa totalmente y su resultado, una tabla temporal, se utiliza como criterio de selección en la consulta superior.
- En una subconsulta correlacionada, la subconsulta anidada depende de una variable que recibe un valor desde la consulta de nivel superior.
- Por tanto, en una subconsulta correlacionada no se evalúa totalmente devolviendo una tabla temporal, sino que para cada valor de la consulta principal se realiza una evaluación de la consulta anidada.

Subconsultas correlacionadas

- Listado de artículos con el precio de venta más caro de cada color

```
SELECT *  
FROM articulos a1  
WHERE art_pv >= ALL  
    (SELECT art_pv  
     FROM articulos a2  
     WHERE a1.art_col = a2.art_col);
```

¡¡Como se trata de la misma tabla, es necesario usar alias (a1, a2)!!

Consulta de existencia

- El operador existencial EXISTS se emplea para realizar consultas de existencia, consultas en que necesitemos utilizar el cuantificador universal “para todo” o la operación de división del álgebra relacional.
- La subconsultas de existencia debe colocarse dentro de una cláusula WHERE y después de la palabra clave EXISTS. La condición se cumple si la consulta anidada da un resultado de una fila por lo menos.

Consulta de existencia

- Listado de clientes que han realizado alguna compra (también podría leerse como *listado de clientes para los que existe alguna compra*)

```
SELECT *  
FROM clientes  
WHERE EXISTS  
    (SELECT *  
     FROM ventas  
     WHERE clt_num = vnt_clt);
```

Referencias

[1] MySQL: <https://dev.mysql.com/doc/refman/8.0/en/func-op-summary-ref.html>