

Funciones y expresiones

Tema 2

Bases de Datos - Grado en Ingeniería Informática
Antonio Balderas Alberico

Comentarios

- Transparencias correspondientes al **capítulo 3** 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.

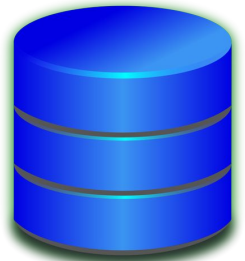
Introducción

- Al trabajar con una base de datos nos encontramos con diferentes tipos de datos
- A estos datos se le pueden aplicar funciones y expresiones
- Estas funciones se pueden aplicar a registros individuales o a grupos de ellos que cumplan una determinada condición



Expresiones

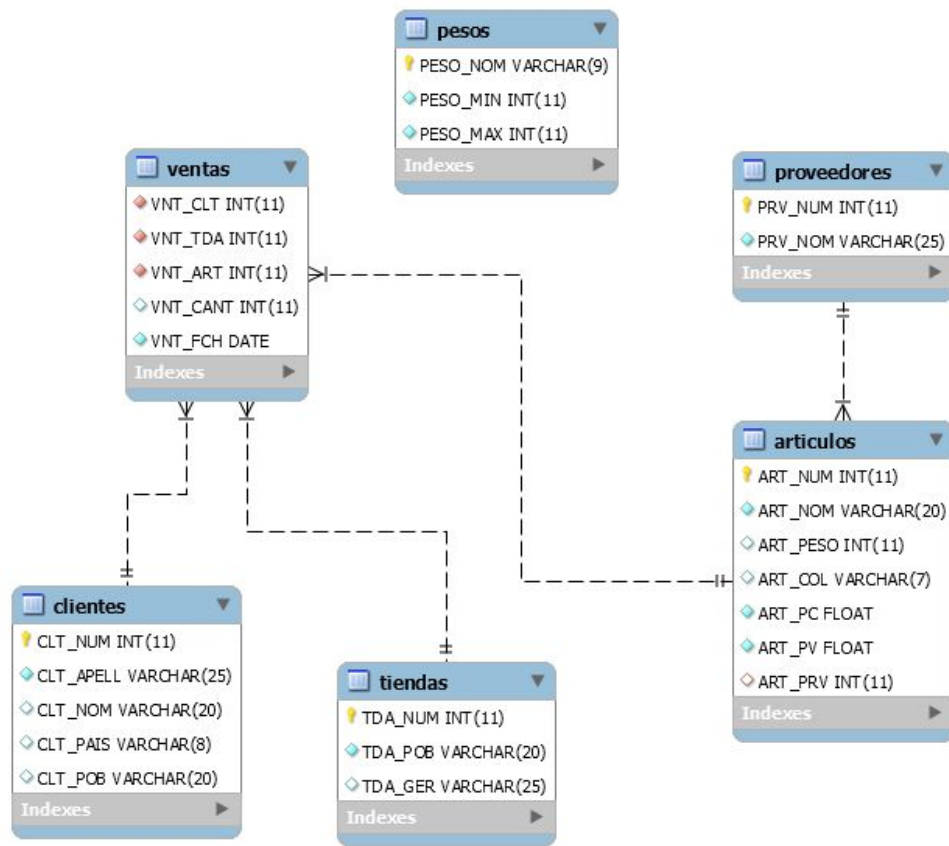
- Nos permiten realizar cálculos numéricos con los datos de la base de datos.
- Se puede combinar el valor de diferentes columnas mediante el uso de operadores aritméticos (+, -, *, /)



Expresiones

EJEMPLO

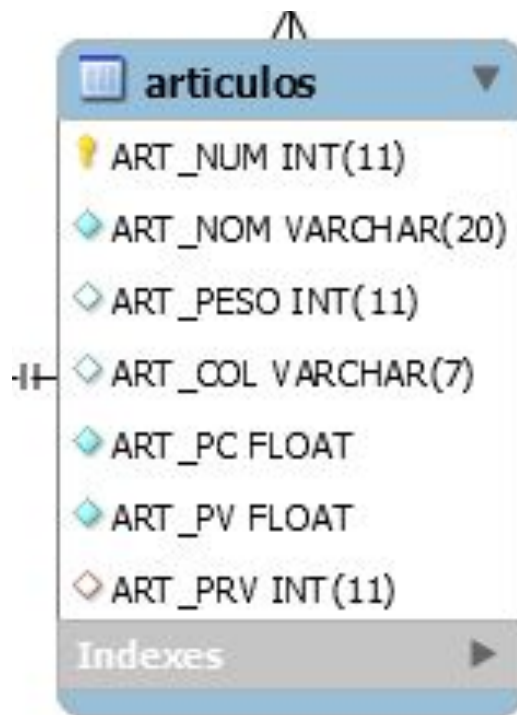
- Volvemos a nuestra base de datos de ventas, clientes, artículos, etc.



Expresiones

EJEMPLO

- Volvemos a nuestra base de datos de ventas, clientes, artículos, etc.
- En la tabla artículos, se almacena el precio de compra de los artículos (art_pc) y el precio de venta (art_pv)



A screenshot of a database management tool showing the structure of a table named 'articulos'. The table has seven columns: ART_NUM (INT(11)), ART_NOM (VARCHAR(20)), ART_PESO (INT(11)), ART_COL (VARCHAR(7)), ART_PC (FLOAT), ART_PV (FLOAT), and ART_PRV (INT(11)). The first four columns have a diamond icon, and the last three have a light blue diamond icon. A primary key icon (a circle with a vertical line) is shown to the left of the ART_COL column. At the bottom, there is a tab labeled 'Indexes' with a right-pointing arrow.

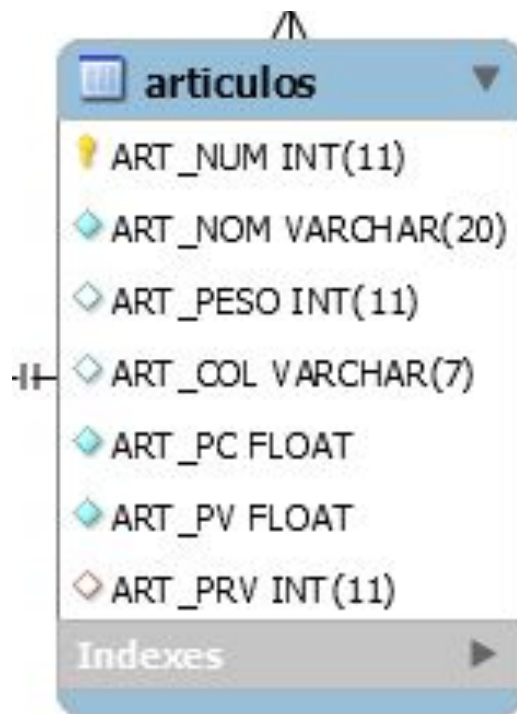
articulos	
ART_NUM	INT(11)
ART_NOM	VARCHAR(20)
ART_PESO	INT(11)
ART_COL	VARCHAR(7)
ART_PC	FLOAT
ART_PV	FLOAT
ART_PRV	INT(11)

Indexes

Expresiones

```
SELECT art_num 'Código de artículo', art_pc  
       'Precio de compra', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Precio de compra	Precio de venta
1	10.00	15.00
2	8.00	12.99
3	10.00	21.50
4	6.00	18.00

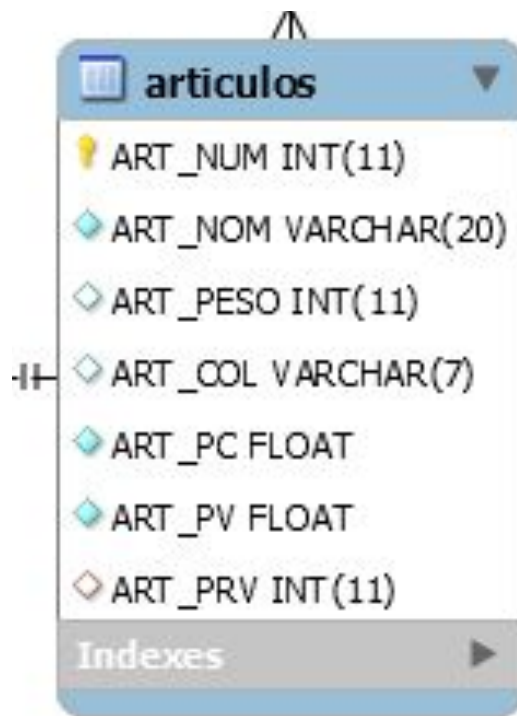


Expresiones

```
SELECT art_num 'Código de artículo', art_pc  
       'Precio de compra', art_pv 'Precio de venta'  
FROM  artículos;
```

Código de artículo	Precio de compra	Precio de venta
1	10.00	15.00
2	8.00	12.99
3	10.00	21.50
4	6.00	18.00

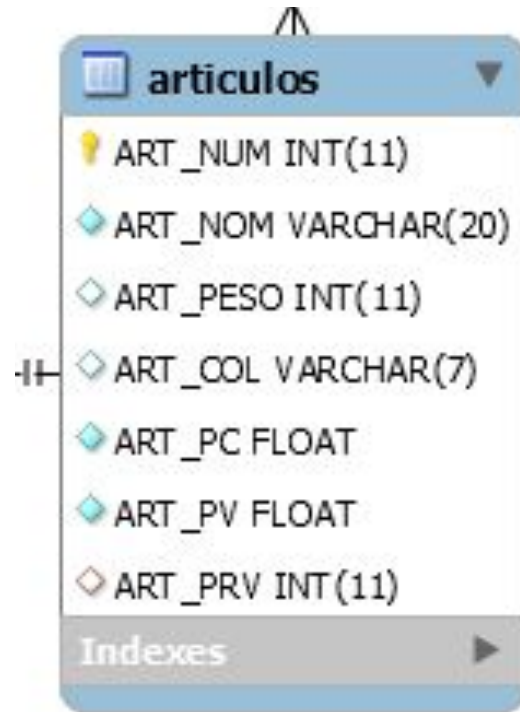
Si tenemos precio de compra y precio de venta, ¿cómo podríamos calcular el beneficio?



Expresiones

```
SELECT art_num 'Código de artículo', art_pv -  
art_pc 'Beneficio' FROM articulos;
```

Código de artículo	Beneficio
1	5.00
2	4.99
3	11.50
4	12.00

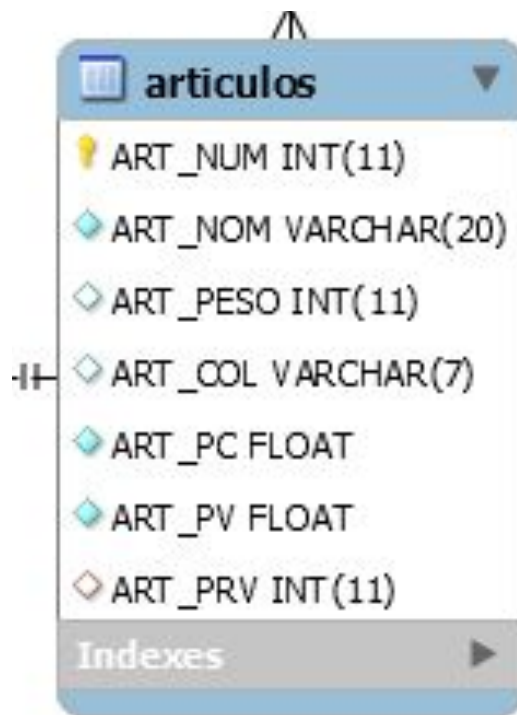


Expresiones

```
SELECT art_num 'Código de artículo', art_pv -  
art_pc 'Beneficio'  
FROM articulos  
WHERE art_pv - art_pc > 10  
ORDER BY 2 DESC;
```

Código de artículo	Beneficio
4	12.00
3	11.50

**Las expresiones también se pueden utilizar en el
WHERE y en el ORDER BY**

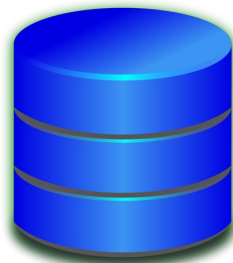


Funciones de registros únicos

Una función de registro único devuelve un único registro como resultado por cada registro de la tabla consultada.

Tipos de funciones de registros únicos:

- Funciones numéricas
- Funciones de caracteres
- Funciones de fechas
- Funciones de conversión
- Otras

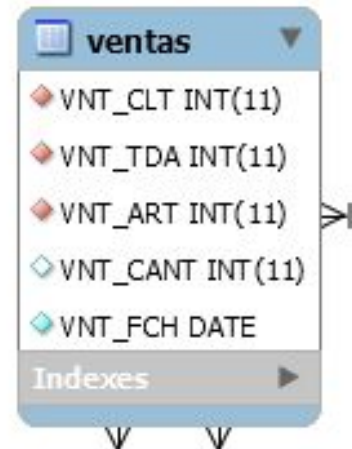


Funciones de registros únicos

Listado que me proporcione las ventas del año 2020

```
SELECT *  
FROM ventas  
WHERE YEAR(vnt_fch) = 2020;
```

Las funciones también se pueden utilizar en el **SELECT**, en el **WHERE** y en el **ORDER BY**

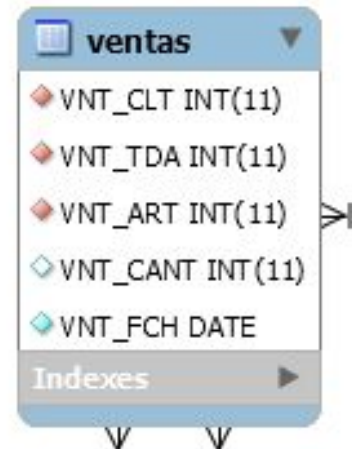


Funciones de registros únicos

Listado que me proporcione las ventas del año 2020

```
SELECT *  
FROM ventas  
WHERE YEAR(vnt_fch) = 2020;
```

Las funciones también se pueden utilizar en el **SELECT**, en el **WHERE** y en el **ORDER BY**



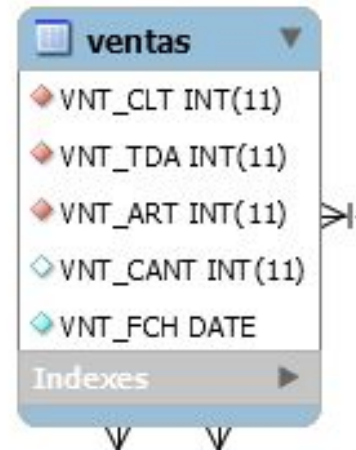
Funciones de registros únicos

Las funciones también se pueden aplicar directamente sobre valores, por ejemplo:

Raíz cuadrada de 9 → **SELECT** SQRT(9);

Número PI → **SELECT** PI();

Función Redondeo → **SELECT** ROUND(PI(),2);



Funciones de registros únicos

CONCAT()	Recibe varias cadenas de caracteres y la devuelve concatenada
CURDATE()	Devuelve la fecha actual
DAYNAME()	Recibe una fecha y devuelve el día de la semana (lunes, martes, etc)
DATEDIFF()	Recibe dos fechas y devuelve la diferencia en días entre ambas
LOWER()	Convierte la cadena completa a minúsculas
PI()	Devuelve el número PI
ROUND()	Redondea un número con tantos decimales como le indiquemos
SQRT()	Devuelve la raíz cuadrada del número recibido
UPPER()	Convierte la cadena completa a mayúsculas



Listado de funciones en MySQL

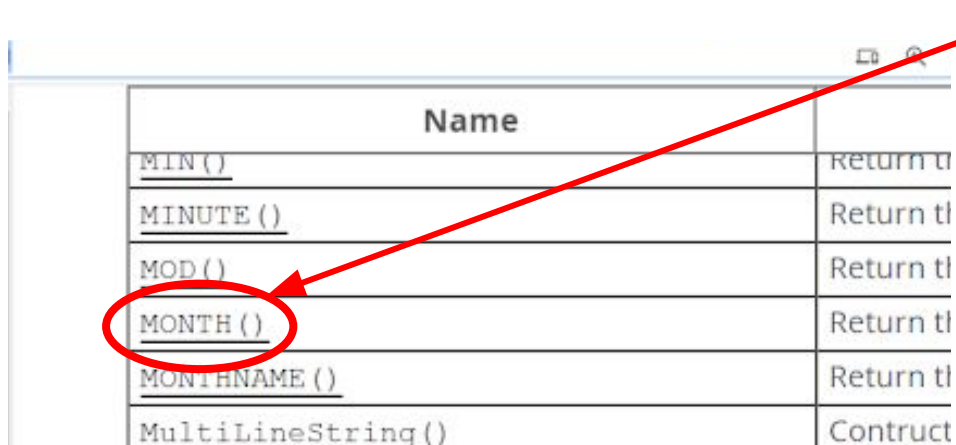
- Listado de funciones disponibles en MySQL:
<https://dev.mysql.com/doc/refman/8.0/en/built-in-function-reference.html>

Name	
<u>MIN()</u>	Return ti
<u>MINUTE()</u>	Return tl
<u>MOD()</u>	Return tl
<u>MONTH()</u>	Return tl
<u>MONTHNAME()</u>	Return tl
MultiLineString()	Construct

Listado de funciones en MySQL

- Listado de funciones disponibles en MySQL:
<https://dev.mysql.com/doc/refman/8.0/en/built-in-function-reference.html>

Si pinchamos sobre una de ellas, por ejemplo month()



A screenshot of a table listing MySQL built-in functions. The table has two columns: 'Name' and a description. The functions listed are MIN(), MINUTE(), MOD(), MONTH(), MONTHNAME(), and MultiLineString(). The 'MONTH()' function is circled in red, and a red arrow points to it from the right.

Name	
<u>MIN()</u>	Return ti
<u>MINUTE()</u>	Return ti
<u>MOD()</u>	Return ti
<u>MONTH()</u>	Return ti
<u>MONTHNAME()</u>	Return ti
MultiLineString()	Construct

Listado de funciones en MySQL

- Listado de funciones disponibles en MySQL:
<https://dev.mysql.com/doc/refman/8.0/en/built-in-function-reference.html>

Se carga un ejemplo de uso y su descripción (en inglés)

- MONTH (date)

Returns the month for *date*, in the range 1 to 12 for January to December, or 0 for dates such as '0000-00-00' or '2008-00-00' that have a zero month part.

```
1  mysql> SELECT MONTH('2008-02-03');  
2      -> 2
```

Funciones y expresiones

Tema 2 (continuación)

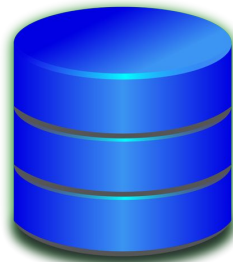
Bases de Datos - Grado en Ingeniería Informática
Antonio Balderas Alberico

Funciones

Una función manipula un conjunto de datos y devuelve un resultado

Tipos de funciones:

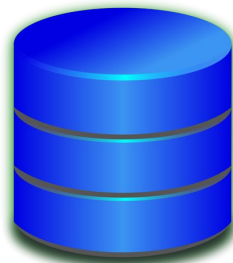
- Funciones de registros únicos
- Funciones de grupos



Funciones de grupo

Las funciones de grupo devuelven un valor asociado a un conjunto de registros ya agrupados. Las funciones de grupos más comunes son:

- **COUNT**: devuelve el número total de registros
- **MIN**: devuelve el mínimo valor del campo que le especifiquemos
- **MAX**: devuelve el máximo valor del campo que le especifiquemos
- **SUM**: devuelve la suma de valores del campo que le especifiquemos
- **AVG**: devuelve la media de los valores del campo que le especifiquemos

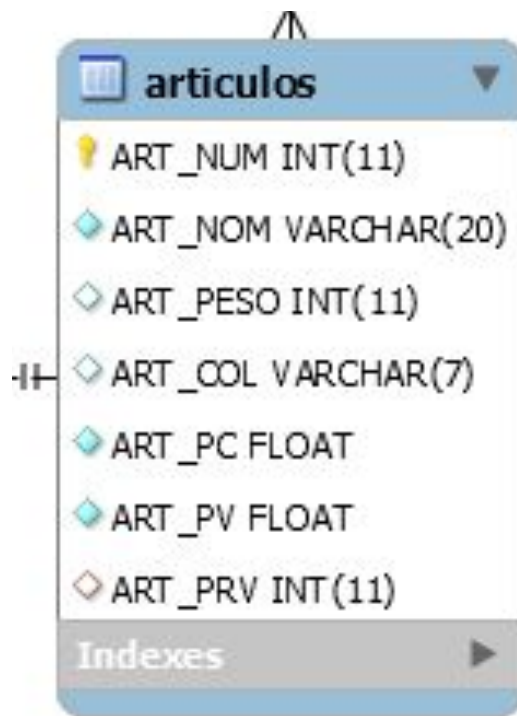


Funciones de grupo

```
SELECT art_num 'Código de artículo', art_pc  
       'Precio de compra', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Precio de compra	Precio de venta
1	10.00	15.00
2	8.00	12.99
3	10.00	21.50
4	6.00	18.00

Por ejemplo ¿Qué funciones de grupo
podríamos aplicar aquí?



Funciones de grupo

```
SELECT art_num 'Código de artículo', art_pc  
       'Precio de compra', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Precio de compra	Precio de venta
1	10.00	15.00
2	8.00	12.99
3	10.00	21.50
4	6.00	18.00

Una función que implique a todo el conjunto de registros:

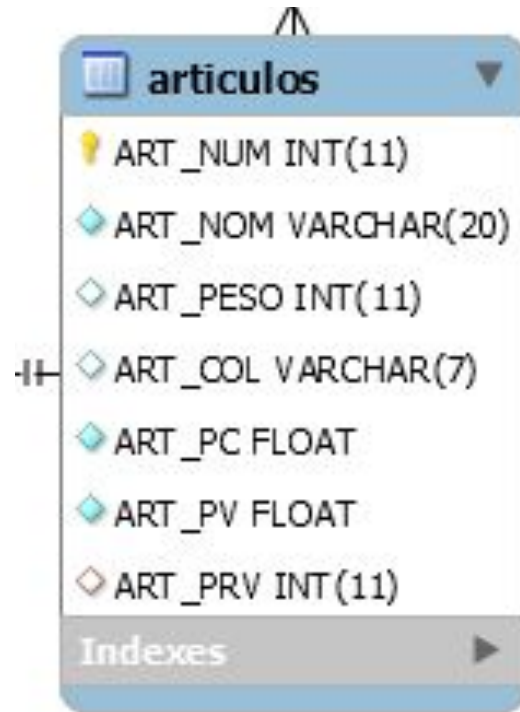
- Media de precios de compra o de venta
- Suma de precios de compra o de venta
- Máximo precio de compra
- Etc.

Por ejemplo ¿Qué funciones de grupo podríamos aplicar aquí?

Funciones de grupo

```
SELECT AVG(art_pc) 'Media precio de compra',  
FROM articulos;
```

Media Precio de compra
8.50



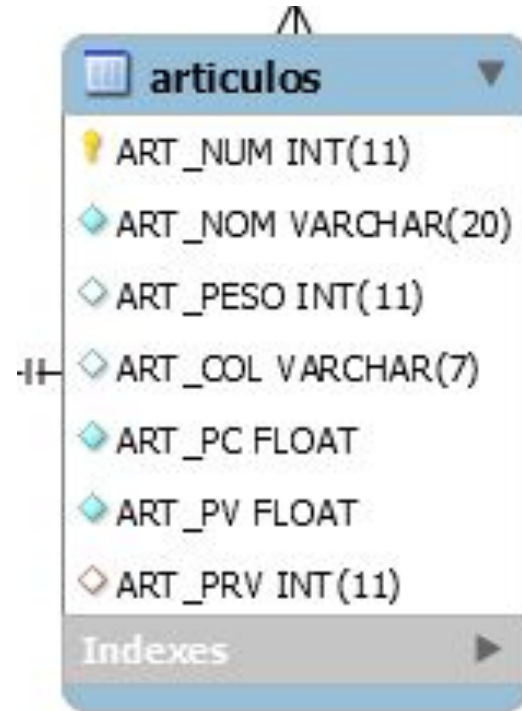
Funciones de grupo

```
SELECT AVG(art_pc) 'Media precio de compra',  
FROM articulos;
```

Media Precio de compra
8.50

```
SELECT MIN(art_pv) 'Mínimo precio de venta',  
FROM articulos;
```

Mínimo precio de venta
12.99



Funciones de grupo

- **GROUP BY** agrupa las filas que tienen los mismos valores en columnas indicadas [2]
- El extracto GROUP BY se utiliza a menudo con funciones de grupo (COUNT, MAX, MIN, SUM, AVG) para agrupar el conjunto de resultados por una o más columnas
- Sintaxis:

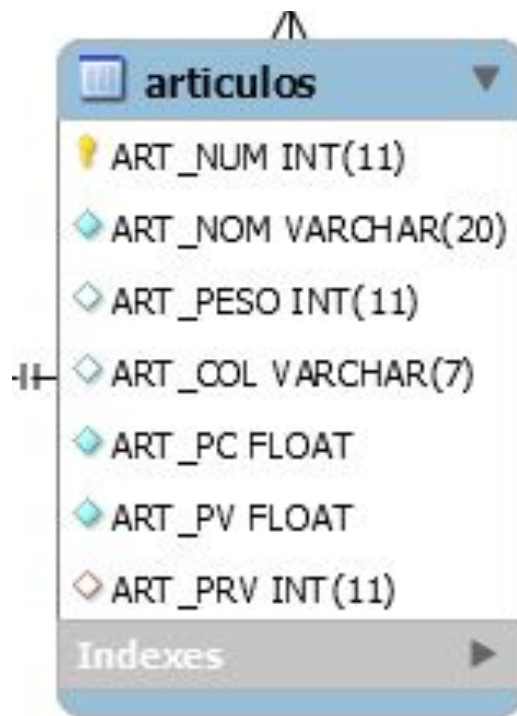
```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
ORDER BY column_name(s);
```

Funciones de grupo

```
SELECT art_num 'Código de artículo', art_col  
'Color', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Color	Precio de venta
1	verde	15.00
2	azul	12.99
3	verde	21.50
4	azul	18.00

En esta consulta vemos que hay 2 artículos verdes

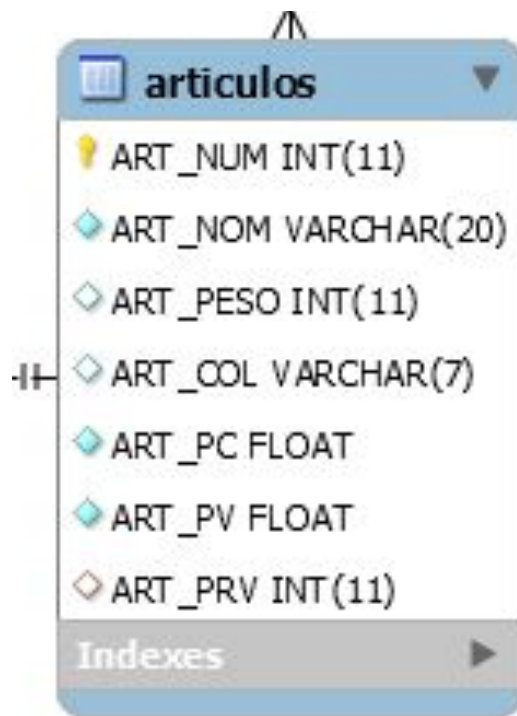


Funciones de grupo

```
SELECT art_num 'Código de artículo', art_col  
'Color', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Color	Precio de venta
1	verde	15.00
2	azul	12.99
3	verde	21.50
4	azul	18.00

En esta consulta vemos que hay 2 artículos
verdes y 2 azules

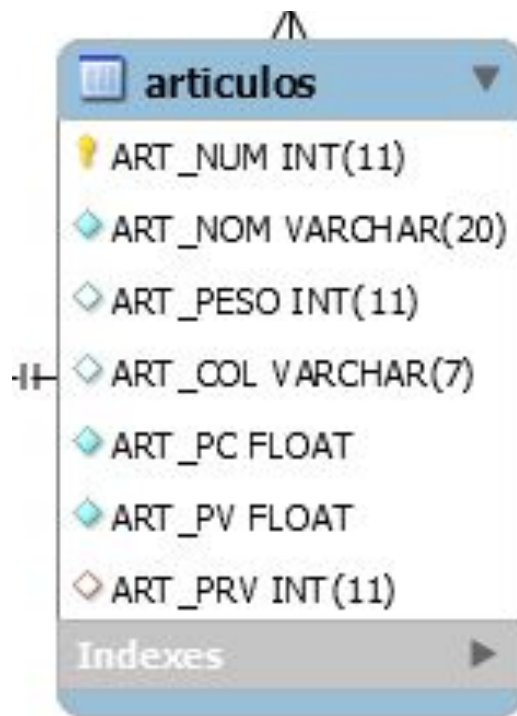


Funciones de grupo

```
SELECT art_num 'Código de artículo', art_col  
'Color', art_pv 'Precio de venta'  
FROM articulos;
```

Código de artículo	Color	Precio de venta
1	verde	15.00
2	azul	12.99
3	verde	21.50
4	azul	18.00

Mediante **GROUP BY**, podríamos sacar la
media por color

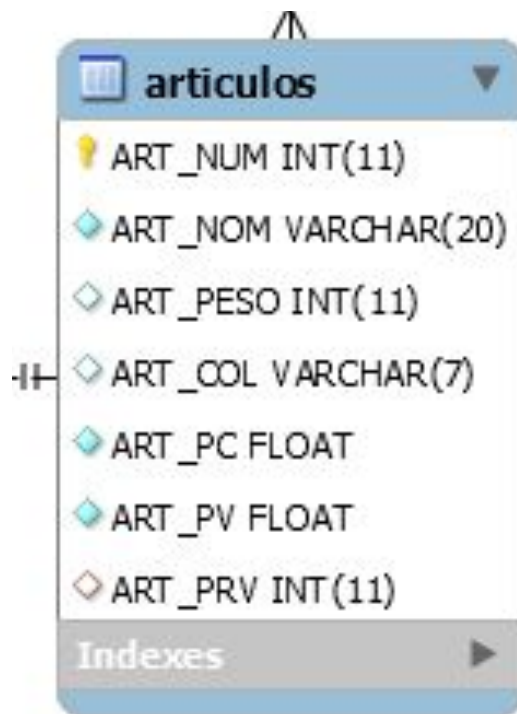


Funciones de grupo

```
SELECT art_col 'Color', AVG(art_pv) 'Precio  
de venta medio'  
FROM articulos  
GROUP BY art_col;
```

Color	Precio de venta medio
verde	18.25
azul	15.49

Mediante HAVING, podríamos eliminar del listado aquellos colores cuyo precio de venta medio no supere un umbral



Funciones de grupo

HAVING: cláusula que permite establecer condiciones a los grupos

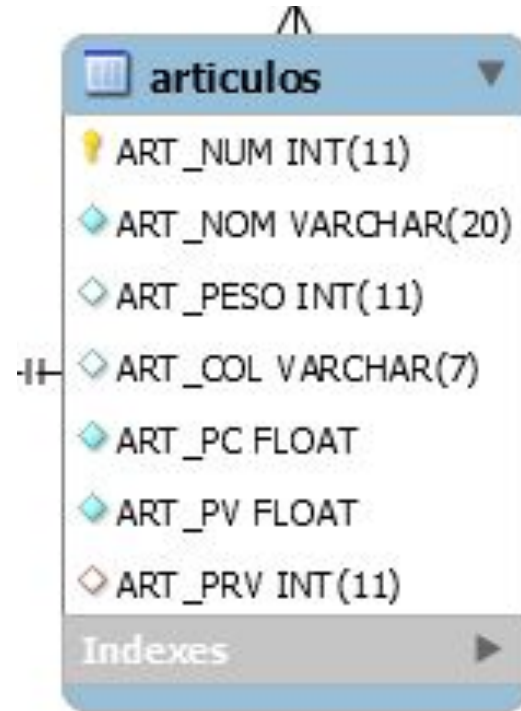
- Sintaxis:

```
SELECT column_name(s)  
FROM table_name  
WHERE condition  
GROUP BY column_name(s)  
HAVING condition  
ORDER BY column_name(s);
```

Funciones de grupo

```
SELECT art_col 'Color', AVG(art_pv) 'Precio  
de venta medio'  
FROM articulos  
GROUP BY art_col  
HAVING AVG(art_pv) > 16;
```

Color	Precio de venta medio
verde	18.25



Funciones de grupo

Importante:

1. En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - Funciones de grupo

Funciones de grupo

Importante:

- En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - Funciones de grupo

```
SELECT art_col 'Color', AVG(art_pv) 'Precio de venta medio'  
FROM articulos  
GROUP BY art_col;
```

Color	Precio de venta medio
verde	18.25
azul	15.49

Funciones de grupo

Importante:

- En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - **Funciones de grupo**

```
SELECT art_col 'Color', AVG(art_pv) 'Precio de venta medio'  
FROM articulos  
GROUP BY art_col;
```

Color	Precio de venta medio
verde	18.25
azul	15.49

Funciones de grupo

Importante:

- En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - Funciones de grupo

```
SELECT art_num, art_col 'Color', AVG(art_pv) 'Precio de venta medio'
FROM articulos
GROUP BY art_col;
```

Esto es un error porque **art_num** no es un atributo del grupo. Para un grupo habrá muchos **art_num**, por lo que esto es un error que cada SGBD gestiona diferente (lancan error o devuelven **art_num** aleatorio)

art_num	Color	Precio de venta medio
???	verde	18.25
???	azul	15.49

Funciones de grupo

Importante:

1. En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - Funciones de grupo
2. El filtrado a nivel de grupo se hace en el HAVING y nivel de registro en el WHERE

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros
HAVING count(*) >= 10

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros
HAVING count(*) >= 10
- Que se muestren los que valen más de 10 euros

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros
HAVING count(*) >= 10
- Que se muestren los que valen más de 10 euros **WHERE** art_pv > 10

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros
HAVING count(*) >= 10
- Que se muestren los que valen más de 10 euros **WHERE** art_pv > 10
- Que se muestren los artículos por colores cuyo precio de venta medio sea mayor de 15.50

Funciones de grupo

En un listado de artículos:

- Que no se muestren los de color rojo **WHERE** art_col <> 'rojo'
- Que no se muestren los colores de los que haya menos de 10 registros
HAVING count(*) >= 10
- Que se muestren los que valen más de 10 euros **WHERE** art_pv > 10
- Que se muestren los artículos por colores cuyo precio de venta medio sea mayor de 15.50 **HAVING** AVG(art_pv) > 15.50

Funciones de grupo

Importante:

1. En el SELECT de una consulta de grupos sólo pueden aparecer:
 - Aquellos campos por los que se ha agrupado
 - Funciones de grupo
2. El filtrado a nivel de grupo se hace en el HAVING y nivel de registro en el WHERE
3. Las funciones de grupo que reciben una columna no tienen en cuenta los registros nulos al realizar el cálculo para esa columna

Funciones de grupo

Las funciones de grupo que reciben una columna no tienen en cuenta los registros nulos al realizar el cálculo para esa columna

art_num	art_col
1	rojo
2	verde
3	NULL
4	verde
5	gris

`count(*) = 5`

`count(art_col) = 4`

`count(distinct art_col) = 3`

Referencias

[1] MySQL: <https://dev.mysql.com/doc/refman/8.0/en/built-in-function-reference.html>

[2] GROUP BY: https://www.w3schools.com/sql/sql_groupby.asp