

CFP
Programador
full-stack

Base de Datos

CFP Programador full-stack

Funciones de Agrupamiento

Funciones de Agrupamiento

Estas funciones operan sobre conjuntos de filas para proporcionar **un resultado por grupo**

- La función COUNT devuelve el número de filas o valores especificados en una consulta
- Las funciones SUM, MAX, MIN y AVG se aplican a un conjunto o multiconjunto de valores numéricos y devuelven, respectivamente, la suma, el valor máximo, el valor mínimo y el promedio de esos valores

Funciones de Agrupamiento

Ejemplos

Obtener el número total de productos cuyo nombre es “fish”

```
SELECT COUNT(*)  
FROM e01_producto  
WHERE nombre LIKE 'fish';
```

count(*)

3

Funciones de Agrupamiento

Ejemplos

Obtener el promedio del precio de los productos cuyo nombre es “fish”

```
SELECT AVG(precio)
FROM e01_producto
WHERE nombre LIKE 'fish';
```

avg(precio)

817.079996
7447916

Funciones de Agrupamiento

Ejemplos

Seleccionar el precio del producto más caro

```
SELECT  
  MAX (precio)  
FROM e01_producto;
```

MAX (precio)

946.390014648437
5

Funciones de Agrupamiento

Ejemplos

Obtener el precio del producto más barato

```
SELECT  
  MIN (precio)  
FROM e01_producto;
```

MIN (precio)

2.14000010490417
5

Funciones de Agrupamiento

- A veces queremos aplicar funciones a subgrupos de filas de una tabla, estando los subgrupos divididos en base a algunos valores de uno o más atributos
- Cada grupo estará compuesto por las filas que tienen el mismo valor para algún(os) atributo(s), denominado(s) atributo(s) de agrupamiento
- Después podemos aplicar la función independientemente a cada grupo

GROUP BY

La cláusula **GROUP BY** especifica los atributos de agrupamiento, que también deben aparecer en la cláusula **SELECT**

- Si se usa la cláusula **GROUP BY** en una sentencia **SELECT**, se dividen las filas de la tabla consultada en grupos

```
SELECT <atributos>, <fun de grupo>  
    (<nombre_columna>)  
FROM <nombre_tabla>  
[WHERE condicion]  
GROUP BY expresión de grupo;
```

GROUP BY

- Se aplica las funciones en la lista `SELECT` a cada grupo de filas y retorna una única fila por cada grupo.

```
SELECT <atributos>, <fun de grupo>  
    (<nombre_columna>)  
FROM <nombre_tabla>  
[WHERE condición]  
GROUP BY expresión de grupo;
```

Todas las columnas de la lista `SELECT`, excepto las funciones de grupo, deben estar en la cláusula `GROUP BY`

GROUP BY

Ejemplo

Listar todos los productos junto con la cantidad que hay de cada uno.

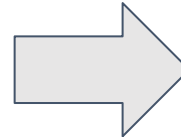
```
SELECT nombre, COUNT (nombre)
FROM e01_producto
GROUP BY nombre;
```

GROUP BY

Ejemplo

```
SELECT nombre,
       COUNT(codigo_producto)
FROM e01_producto
GROUP BY nombre;
```

codigo_producto	nombre	
11	bakery	...
25	bag	...
38	bag	...
...



nombre	COUNT
drugstore	4
aisle	1
bag	2
bakery	2
banknote	1
barcode	2

HAVING

Se puede anexar la cláusula HAVING para restringir grupos

- Las filas se agrupan por la/s columnas especificada/s
- Se aplica la función de grupo
- Se muestran los grupos que satisfacen la cláusula

```
SELECT <lista_atributos>  
FROM <nombre_tabla>  
[WHERE condición]  
GROUP BY <nombre_columna>  
HAVING where_condición;
```

HAVING siempre debe ir acompañado previamente de un GROUP BY

HAVING

Ejemplo

- Listar todas las marcas, cuyos productos tengan un stock inferior a 50 unidades

```
SELECT marca, SUM(stock)
FROM e01_producto
GROUP BY marca
HAVING SUM(stock) < 50;
```

marca	SUM (stock)
At Auctor PC	46
Cras Ltd	44
Fringilla ...	12
Maecenas	39
...	

Restricciones

- **NO** se puede utilizar la cláusula WHERE para restringir grupos
- Se debe utilizar la cláusula HAVING para restringir grupos
- **NO** se pueden utilizar funciones de grupo en la cláusula WHERE

ORDER BY

SQL permite ordenar las filas del resultado de una consulta por los valores de uno o más atributos, utilizando la cláusula ORDER BY. El orden predeterminado es el **ascendente**

- DESC podemos ver el resultado ordenado descendentemente
- ASC permite especificar explícitamente el orden ascendente

```
SELECT <lista_atributos>  
FROM <nombre_tabla>  
...  
ORDER BY <nombre_columna> [ASC | DESC];
```


ORDER BY

Ejemplo

Listar los datos de los clientes ordenados por apellido y nombre

```
SELECT *  
FROM e01_cliente  
ORDER BY apellido, nombre ASC;
```

cod	nombre	apellido	...
-----	--------	----------	-----

1	Xerxes	Hale	...
---	--------	------	-----

2	Brent	Leblanc	...
---	-------	---------	-----

3	Kasper	Shannon	...
---	--------	---------	-----

...
-----	-----	-----	-----



cod	nombre	apellido	...
-----	--------	----------	-----

19	Merrill	Allen	...
----	---------	-------	-----

11	Lane	Alvarado	...
----	------	----------	-----

52	Wanda	Baker	...
----	-------	-------	-----

88	Francis	Barker	...
----	---------	--------	-----

89	Ethan	Berger	...
----	-------	--------	-----

...
-----	-----	-----	-----

LIMIT

Se puede utilizar para restringir el número de filas que puede retornar una sentencia

SELECT

- Puede tomar dos argumentos, el primer argumento especifica el desplazamiento de la primera fila a devolver, y el segundo especifica el número máximo de filas a devolver. El desplazamiento de la fila inicial es 0 (no 1)

```
SELECT <lista_atributos>  
FROM <nombre_tabla>  
...  
LIMIT <numero>;
```

LIMIT

Ejemplo

Listar los primeros 3 productos más caros

```
SELECT *  
FROM e01_producto  
ORDER BY precio DESC  
LIMIT 3;
```

cod	marca	nombre	descripción	precio
20	Auctor Corporation	half price	...	946.39
12	Aptent Taciti Incorporated	fish	...	943.44
32	Montes Nascetur Incorporated	cosmetics	...	926.6

Estructura completa de la sentencia SELECT

```
SELECT [ALL | DISTINCT | DISTINCTROW]
      lista_atributos
FROM nombre_tabla
[WHERE condición]
[GROUP BY {nombre_columna | expresión}
  [ASC | DESC], ...]
[WITH ROLLUP]] [HAVING where_condicion]
[ORDER BY {nombre_columna | expr |
  position}
  [ASC | DESC], ...]
[LIMIT {[offset,] row_count | row_count
  OFFSET offset}]
```

Base de Datos

CFP
Programador
full-stack

Funciones de Agrupamiento

Funciones de Agrupamiento

1. Obtener el número total de clientes que se encuentran registrados en la base de datos
2. Listar el precio promedio de cada marca
3. Listar el nombre junto con el precio promedio de los 10 primeros productos ordenados alfabéticamente
4. Listar lo que gastó cada cliente, mostrando el número de cliente y la suma total
5. Listar las marcas cuyo promedio de precios sea mayor a 600\$

Base de Datos

CFP Programador full-stack

Funciones de Agrupamiento

Funciones de Agrupamiento

Repaso

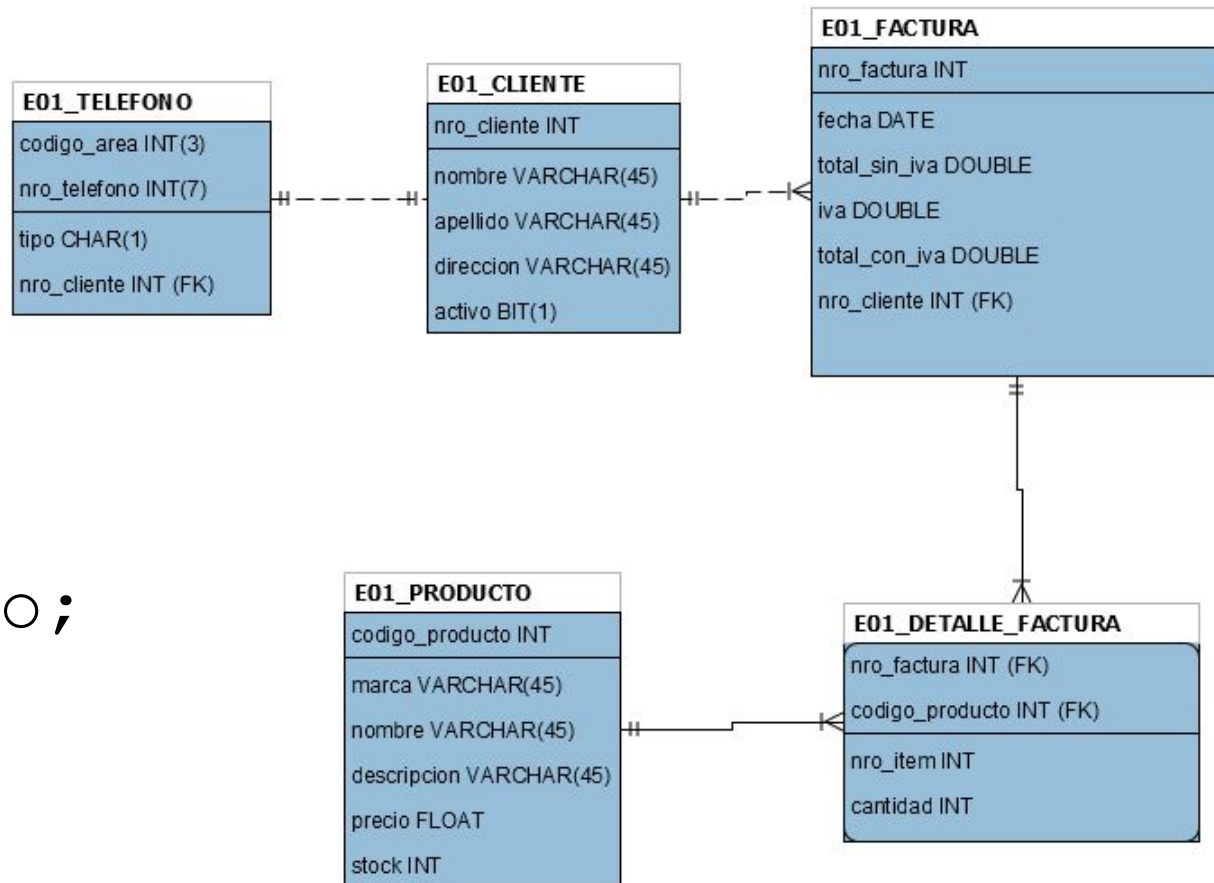
Son funciones que operan sobre conjuntos de filas para proporcionar un resultado por grupo.

Operador	Descripción
SUM ()	Sumatoria de una columna especificada.
AVG ()	Promedio de una columna especificada.
MAX ()	Máximo de una columna especificada.
MIN ()	Mínimo de una columna especificada.
COUNT ()	Cantidad de filas de una columna especificada.

Funciones de Agrupamiento

Ejemplo

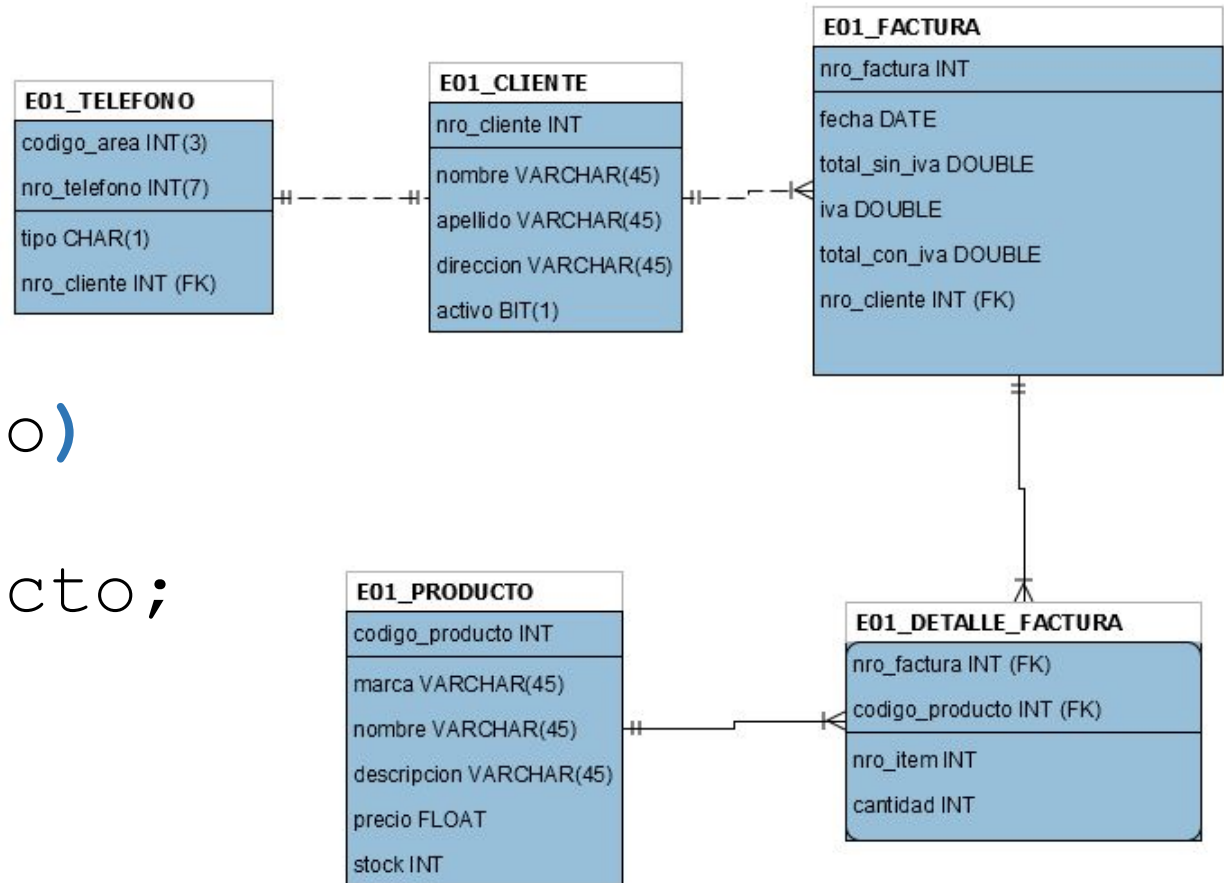
```
SELECT  
    SUM(precio)  
FROM  
    e01_producto;
```



Funciones de Agrupamiento

Ejemplo

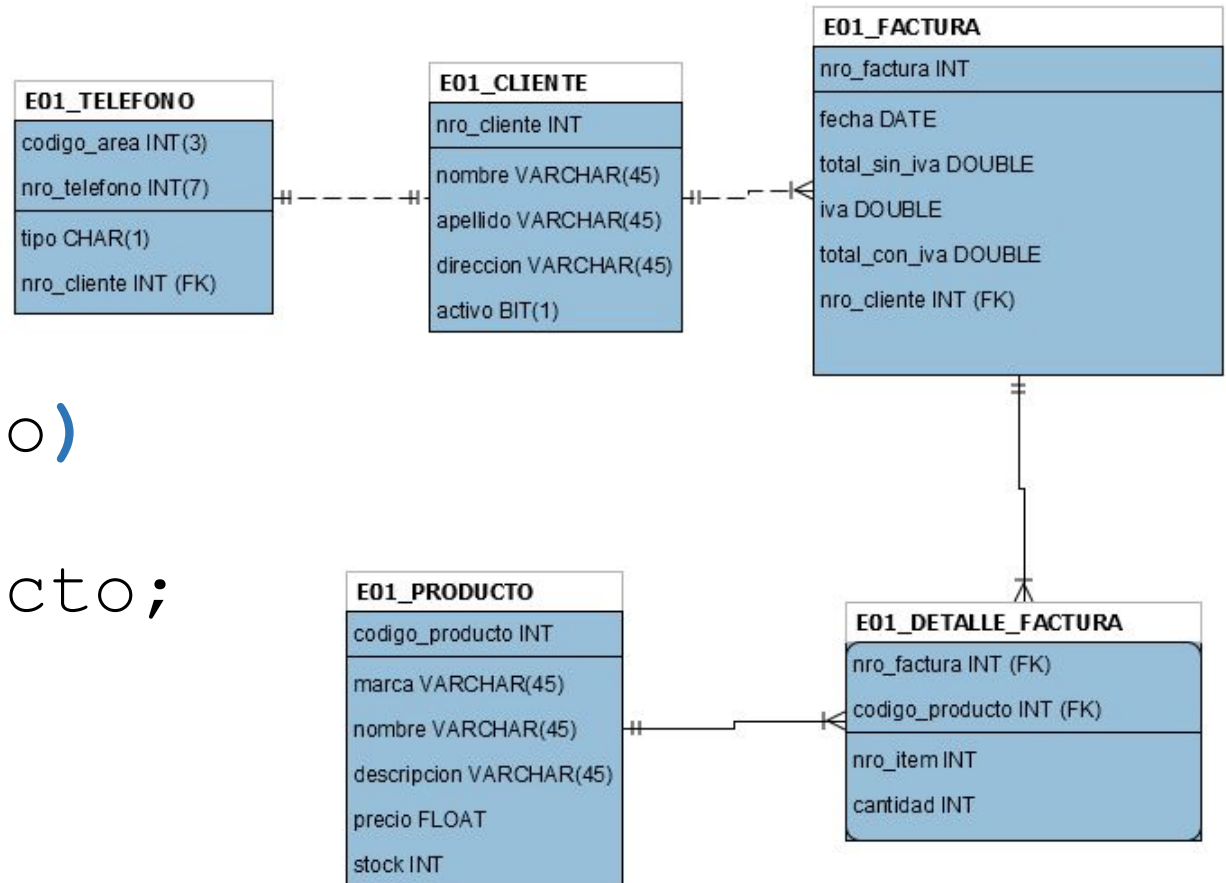
```
SELECT  
    AVG (precio)  
FROM  
    e01_producto;
```



Funciones de Agrupamiento

Ejemplos

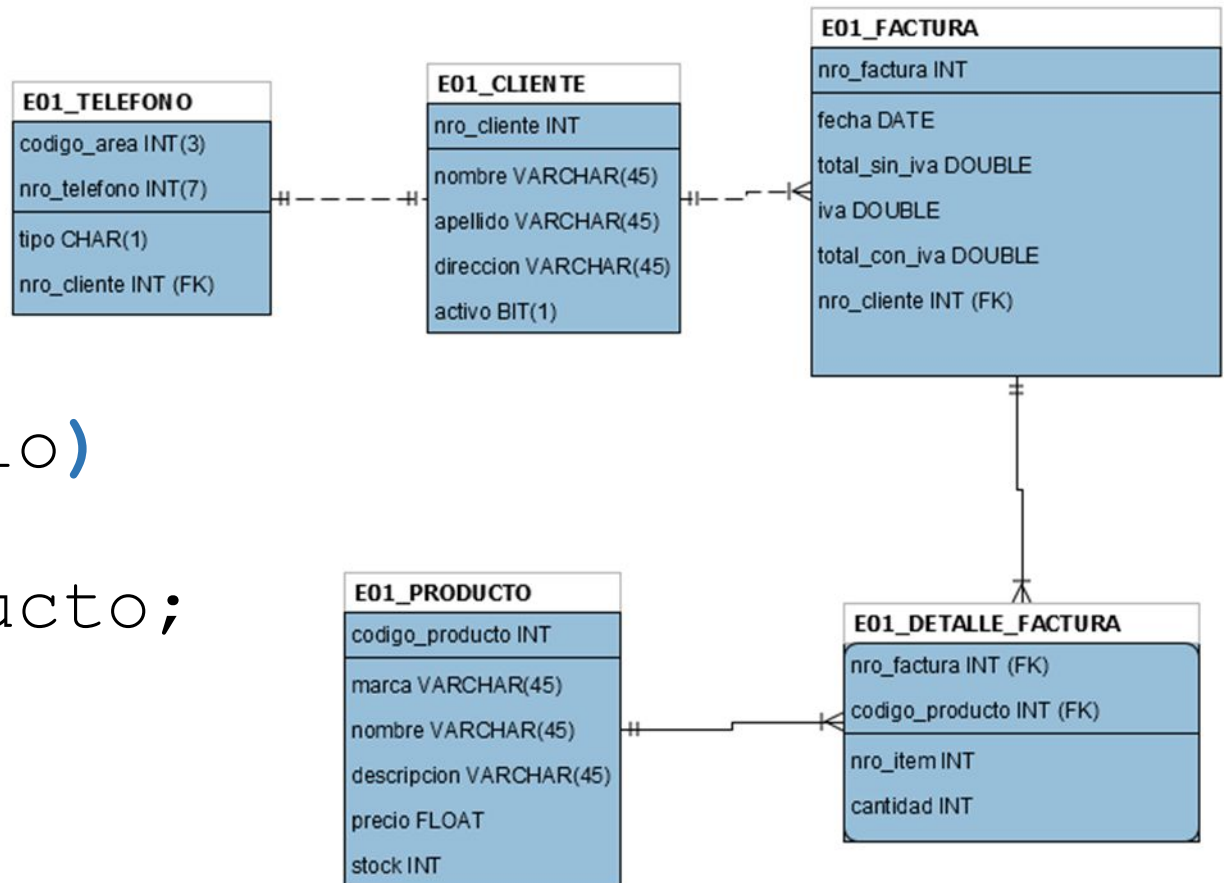
```
SELECT
    MAX (precio)
FROM
    e01_producto;
```



Funciones de Agrupamiento

Ejemplos

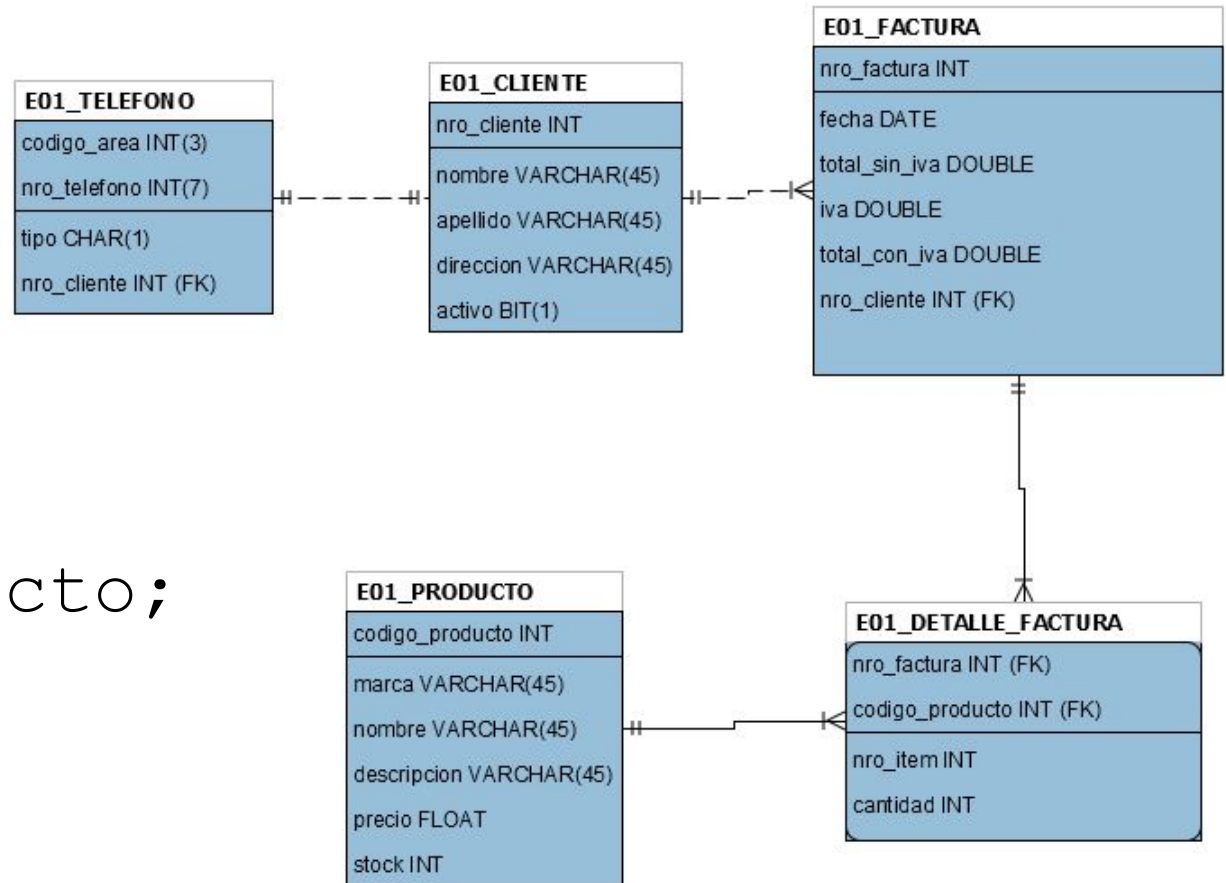
```
SELECT  
    MIN(precio)  
FROM  
    e01_producto;
```



Funciones de Agrupamiento

Ejemplos

```
SELECT  
  COUNT (*)  
FROM  
  e01_producto;
```



Funciones de Agrupamiento

Repaso

- La cláusula GROUP BY se usa para especificar los atributos de agrupamiento
- Se puede anexar la cláusula HAVING para restringir grupos.

Ejemplo:

```
SELECT marca, SUM(stock)
FROM e01_product
GROUP BY marca
HAVING SUM(stock) < 50;
```

Funciones de Agrupamiento

Repaso

Otras funciones que podemos utilizar son:

- LIMIT para restringir el número de filas.
- La cláusula ORDER BY se usa para establecer la columna por cual serán ordenadas las tuplas.

```
SELECT *  
FROM e01_producto  
ORDER BY precio DESC  
LIMIT 3;
```

Sintaxis

Repaso

```
SELECT [ALL | DISTINCT | DISTINCTROW]  
  lista_atributos  
FROM nombre_tabla  
[WHERE condición]  
[GROUP BY {nombre_columna | expresión}  
  [ASC | DESC], ...  
[WITH ROLLUP]] [HAVING where_condicion]  
[ORDER BY {nombre_columna | expr |  
  position}  
[ASC | DESC], ...]  
[LIMIT {[offset,] row_count | row_count  
  OFFSET offset}]
```