

# Módulo de adaptación

## Master en Business Intelligence y Big Data

**PROFESOR/A**  
Antonio Sarasa Cabezuelo

# Operaciones sobre tablas

# Funciones de Agregación

- Las funciones de agregación son funciones que permiten realizar operaciones sobre los datos de una columna. Algunas funciones son las siguientes:

Funciones de agregación	
Función	Descripción
COUNT	Nos da el número total de filas seleccionadas
SUM	Suma los valores de una columna
MIN	Nos da el valor mínimo de una columna
MAX	Nos da el valor máximo de una columna
AVG	Calcula el valor medio de una columna

- En general, las funciones de agregación se aplican a una columna, excepto COUNT que se aplica a todas las columnas de las tablas seleccionadas. Se indica como COUNT (\*).

# Funciones de Agregación

- Sin embargo si se especifica COUNT(distinct column), entonces sólo contará los valores que no nulos ni repetidos, y se especifica COUNT(column), sólo contaría los valores que no nulos.

# Funciones de agregación

- Por ejemplo si se quieren contar el número de clientes de la tabla clientes cuya ciudad es “Madrid”:

```
SELECT COUNT(*) AS numero_clie FROM  
clientes WHERE ciudad = 'Madrid';
```

# Agrupación de filas

- Al realizar una consulta, las filas se pueden agrupar de la siguiente manera:

```
SELECT nombre_columnas_a seleccionar  
FROM tabla_a_consultar [WHERE condiciones]  
GROUP BY columnas_según_las_cuales_se_quiere_agrupar  
[HAVING condiciones_por_grupos]  
[ORDER BY columna_ordenación [DESC] [, columna [DESC]...]];
```

# Agrupación de filas

- La cláusula **GROUP BY** permite agrupar las filas según las columnas indicadas, excepto aquellas afectadas por funciones de agregación.
- La cláusula **HAVING** especifica las condiciones para recuperar grupos de filas.

# Agrupación de filas

- Por ejemplo si se quiere conocer el importe total de los proyectos agrupados por clientes:

```
SELECT código_cliente, SUM(precio) AS importe FROM  
clientes  
GROUP BY codigo_cliente;
```

- Y si solo queremos aquellos clientes con un importe facturado mayor de 10000 euros

```
SELECT código_cliente FROM clientes  
GROUP BY codigo_cliente  
HAVING SUM(precio)>10000
```



# Vistas

- Una vista es una tabla ficticia(no existen como un conjunto de valores almacenados en la base de datos) que se construye a partir de una consulta a una tabla real. Para definir una vista se usa la siguiente sintaxis:

```
CREATE VIEW nombre_vista [(lista_columnas)] AS (consulta)  
[WITH CHECK OPTION];
```

# Vistas

- Donde se indica el nombre de la vista, a continuación se pueden especificar los nombres de las columnas de la vista, se define la consulta que construirá la vista, y se puede añadir la clausula “with check option” para evitar inserciones o actualizaciones excepto en los registros en que la cláusula WHERE de la consulta se evalúe como true.

# Vistas

- Para borrar una vista se utiliza la sentencia **DROP VIEW**:

**DROP VIEW** nombre\_vista (**RESTRICT** | **CASCADE**);

- donde:
  - La opción **RESTRICT** indica que la vista no se borrará si está referenciada,
  - La opción **CASCADE** indica que todo lo que referencie a la vista se borrará con ésta.

# Vistas

Para ilustrar las vistas, se van a considerar las siguientes tablas:

– Tabla clientes:

clientes					
codigo_cli	nombre_cli	nif	dirección	ciudad	teléfono
10	Carrefour	38.567.893-C	Gran vía 11	Madrid	NULL
20	El Corte Inglés	38.123.898-E	Plaza de España 22	Zaragoza	978 45 56 78
30	Mercadona	36.432.127-A	Begoña, 33	Bilbao	940 34 56 90

– Tabla pedidos:

pedidos				
código_pedido	precio	fecha_pedido	fecha_entrega	codigo_cliente
1	1,0E+6	1-1-98	1-1-99	10
2	2,0E+6	1-10-96	31-3-98	10
3	1,0E+6	10-2-98	1-2-99	20

# Vistas

- Si se quiere crear una vista que indique para cada cliente el número de pedidos que tiene encargados el cliente, se definiría la vista:

```
CREATE VIEW pedidos_por_cliente (codigo_cli,  
num_pedidos) AS  
(SELECT c.codigo_cli, COUNT(*) FROM pedidos p,  
clientes c WHERE p.codigo_cliente = c.codigo_cli  
GROUP BY c.codigo_cli);
```

Y se obtendría la vista:

pedidos_por_clientes	
codigo_cli	num pedidos
10	2
20	1
30	1