

A3. Consultas que generan valores de Resumen.

Índice.

1.	Consultas de Resumen.....	2
2.	Formato de una Consulta de Resumen.....	3
3.	Tipos de Consultas de Resumen.....	5
3.1.	Funciones asociadas a las columnas.....	6
3.2.	Agrupación de registros.....	9
3.3.	Filtrado de resultado.....	10

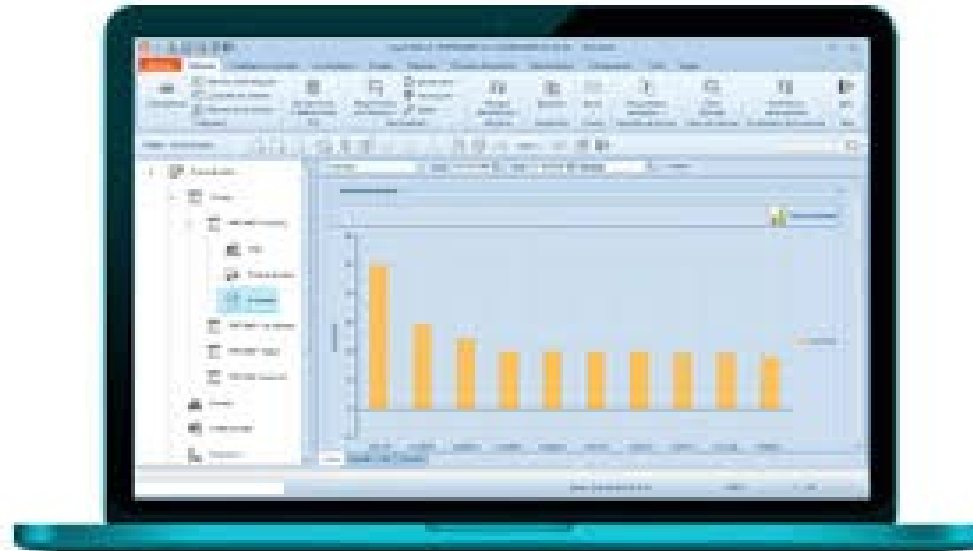
A3. Consultas que generan valores de Resumen.

1. Consultas de Resumen.

SQL permite la realización de Consultas complejas que realizan el resumen de información y proporcionen cálculos entre varios conjuntos de registros.

Las **funciones de columna** se encargan de obtener dicha información resumida, transformando un conjunto de registros complejo en una información simple, cuyo resultado es un mero cálculo.

Las funciones de columna

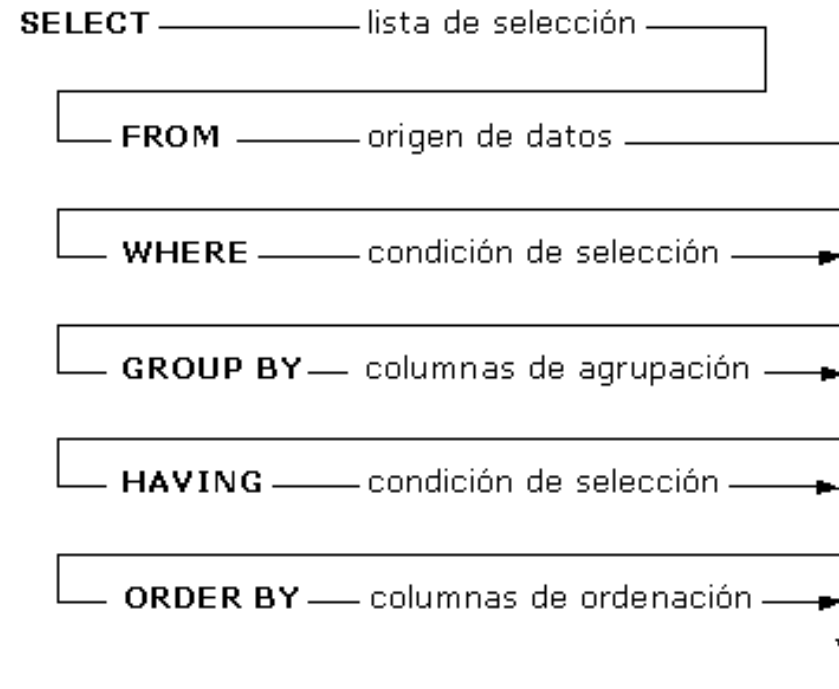


A3. Consultas que generan valores de Resumen.

2. Formato de una Consulta de Resumen.

El formato de una Consulta de Resumen es el siguiente:

```
SELECT _____ lista de selección _____  
FROM _____ origen de datos _____  
WHERE _____ condición de selección _____  
GROUP BY _____ columnas de agrupación _____  
HAVING _____ condición de selección _____  
ORDER BY _____ columnas de ordenación _____  
;
```



Las cláusulas GROUP BY y HAVING sólo se pueden escribir en este tipo de Consultas y deben estar entre las cláusulas WHERE Y ORDER BY.

A3. Consultas que generan valores de Resumen.

3. Tipos de Consultas de Resumen.

Las Consultas de Resumen pueden ser de los siguientes tipos:

- Asociadas a las columnas.
- Agrupaciones de registros.
- Filtrado de resultados.

idealista

Evolución del salario mínimo en España

Año	SMI Mensual	IPC	SMI Mensual	Índice	SMI Anual t.r.	SMI Anual	Salario medio	SMI/Salario Medio
2007	570,60	100,00	570,60	100,00	7.988,40	7.988,40	21.989	36,33
2008	600,00	101,40	591,72	103,70	8.284,02	8.400,00	23.252	36,13
2009	624,00	102,21	610,50	106,99	8.547,01	8.736,00	24.164	36,15
2010	633,30	105,28	601,55	105,42	8.421,74	8.866,20	24.786	35,77
2011	641,40	107,80	594,97	104,27	8.329,55	8.979,60	25.515	35,19
2012	641,40	110,93	578,20	101,33	8.094,80	8.979,60	25.894	34,68
2013	645,30	111,26	579,98	101,64	8.119,66	9.034,20	26.027	34,71
2014	645,30	110,15	585,83	102,67	8.201,67	9.034,20	26.191	34,49
2015	648,60	110,15	588,83	103,19	8.243,62	9.080,40	26.475	34,30
2016	655,20	111,91	585,45	102,60	8.196,36	9.172,80	26.449	34,68
2017	707,70	113,16	625,41	109,61	8.755,74	9.907,80	26.550	37,32
2018	735,90	114,50	642,73	112,64	8.998,15	10.302,60	26.922	38,27
2019	900,00	115,40	779,90	136,68	10.918,60	12.600,00	27.537	45,76
2020	950,00	114,82	827,38	145,00	11.583,35	13.300,00	27.537	48,30

Fuente:INE

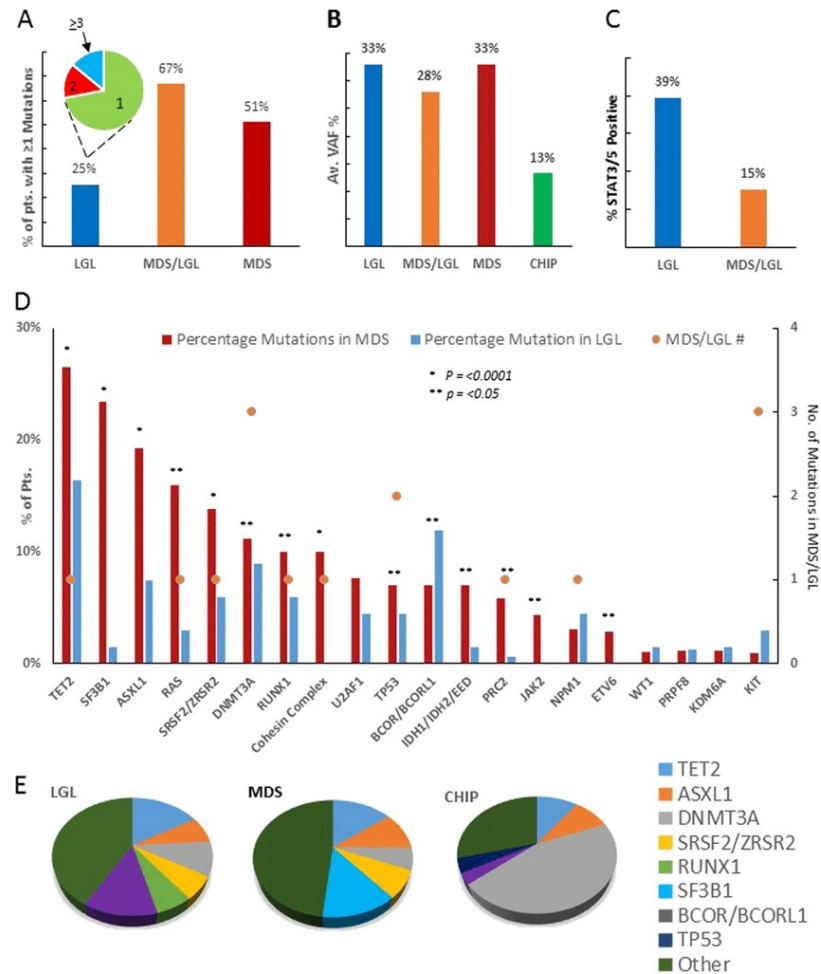
A3. Consultas que generan valores de Resumen.

3.1. Funciones asociadas a las columnas.

Las funciones asociadas a las columnas se pueden utilizar con las cláusulas SELECT, WHERE y ORDER BY.

Las funciones asociadas a las columnas se agrupan en:

- Funciones con cadenas
- Funciones de fecha y hora
- Funciones matemáticas



A3. Consultas que generan valores de Resumen.

3.1. Funciones asociadas a las columnas.

- **Funciones con cadenas**

CONCAT	Concatena cadenas
CONCAT_WS	Concatena cadenas con un separador.
LOWER	Devuelve una cadena en minúscula.
UPPER	Devuelve una cadena en mayúscula.
SUBSTR	Devuelve una subcadena.
LEFT	Devuelve los caracteres de una cadena, empezando por la izquierda.
RIGHT	Devuelve los caracteres de una cadena, empezando por la derecha.
CHAR_LENGTH	Devuelve el numero de caracteres de una cadena.
LENGTH	Devuelve el número de bytes que ocupa una cadena.
REVERSE	Devuelve una cadena invirtiendo el orden de sus caracteres.
LTRIM	Elimina los espacios en blanco que haya al inicio de una cadena.
RTRIM	Elimina los espacios en blanco que haya al final de una cadena.
TRIM	Elimina los espacios en blanco que haya al inicio y al final de una cadena.
REPLACE	Reemplaza un carácter dentro de una cadena.

A3. Consultas que generan valores de Resumen.

3.1. Funciones asociadas a las columnas.

- **Funciones de fecha y hora**

NOW	Devuelve la fecha y la hora actual.
CURTIME	Devuelve la hora actual.
ADDDATE	Suma un número de días a una fecha y calcula la nueva fecha.
DATE_FORMAT	Permite dar formato a las fechas.
DATEDIFF	Calcula el número de días entre dos fechas.
YEAR	Devuelve al año de una fecha.
MONTH	Devuelve el mes de una fecha.
MONTHNAME	Devuelve el nombre del mes de una fecha.
DAY	Devuelve el día de una fecha.
DAYNAME	Devuelve el nombre del día de una fecha.
HOURL	Devuelve las horas de un valor de tipo DATETIME.
MINUTE	Devuelve los minutos de un valor de tipo DATETIME.
SECOND	Devuelve los segundos de un valor de tipo DATETIME.

A3. Consultas que generan valores de Resumen.

3.1. Funciones asociadas a las columnas.

- **Funciones matemáticas**

ABS	Devuelve el valor absoluto.
POW	Devuelve el primer valor elevado al segundo.
SQRT	Devuelve la raíz cuadrada.
PI	Devuelve el valor del número PI.
ROUND	Redondea un valor numérico.
TRUNCATE	Trunca (sesga, parte) un valor numérico.
CEIL	Devuelve el entero inmediatamente superior o igual.
FLOOR	Devuelve el entero inmediatamente inferior o igual.
MOD	Devuelve el resto de una división.
SUM	Devuelve la suma de unos valores.
AVG	Devuelve la media de unos valores.
MIN	Devuelve el valor mínimo de unos valores.
MAX	Devuelve el valor máximo de unos valores.
COUNT	Devuelve el número de elementos de una lista de valores.

A3. Consultas que generan valores de Resumen.

3.2. Agrupación de registros.

La **agrupación de registros** es un tipo de agrupaciones de resumen consistente en crear conjuntos de registros cuyas columnas tengan valores iguales y se realiza con la cláusula **GROUP BY**.

La sintaxis que se debe emplear para la agrupación de registros es la siguiente:

```
SELECT [DISTINCT] expresión_select [,expresión_select] ...  
[FROM tabla]  
[WHERE filtro]  
[GROUP BY expresión [,expresión] ...
```

Si deseamos mostrar el número de jugadores de la NBA por equipos:

```
SELECT nombre_equipo, count(*) AS 'Jugadores' FROM nba.jugadores  
GROUP BY nombre_equipo;
```

A3. Consultas que generan valores de Resumen.

3.3. Filtrado de resultado.

El **filtrado de resultado** es un filtro de grupo realizado con la cláusula **HAVING**, debido a que la cláusula WHERE se utiliza previamente para agrupar los registros.

La sintaxis que se debe emplear para filtrar resultados calculados mediante agrupaciones es:

```
SELECT [DISTINCT] expresión_select [,expresión_select] ...  
[FROM tabla]  
[WHERE filtro]  
[GROUP BY expresión [,expresión] ...]  
[HAVING filtro_grupos]
```

La cláusula HAVING aplica los mismos filtros que la cláusula WHERE.

Si deseamos mostrar los equipos cuyos jugadores pesen de media más de 228 libras:

```
SELECT nombre_equipo, avg(peso) FROM nba.jugadores GROUP BY nombre_equipo  
HAVING avg(peso) > 228  
ORDER BY avg(peso);
```