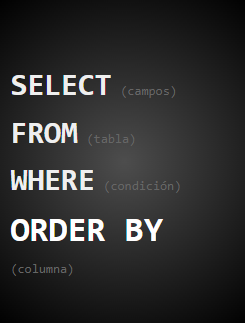
***Clase 04. Sublenguajes SQL***

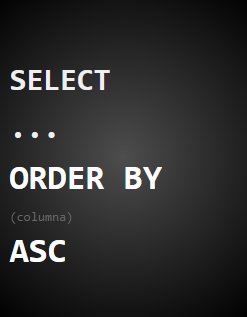
**Ordenamiento de datos**

* Order by

LA SENTENCIA ORDER BY

Así como podemos consultar, seleccionar qué mostrar y cuáles datos filtrar, SQL nos brinda también la posibilidad de ordenar la información. Para el ordenamiento de la misma, debemos recurrir a la sentencia ORDER BY.

Podemos especificar una o más columnas, separando las mismas por una coma, para que el resultado SQL muestre los datos ordenados de acuerdo a nuestro criterio o necesidad.

Existen dos tipos de ordenamiento (de uso opcional):

* ASC
* DESC

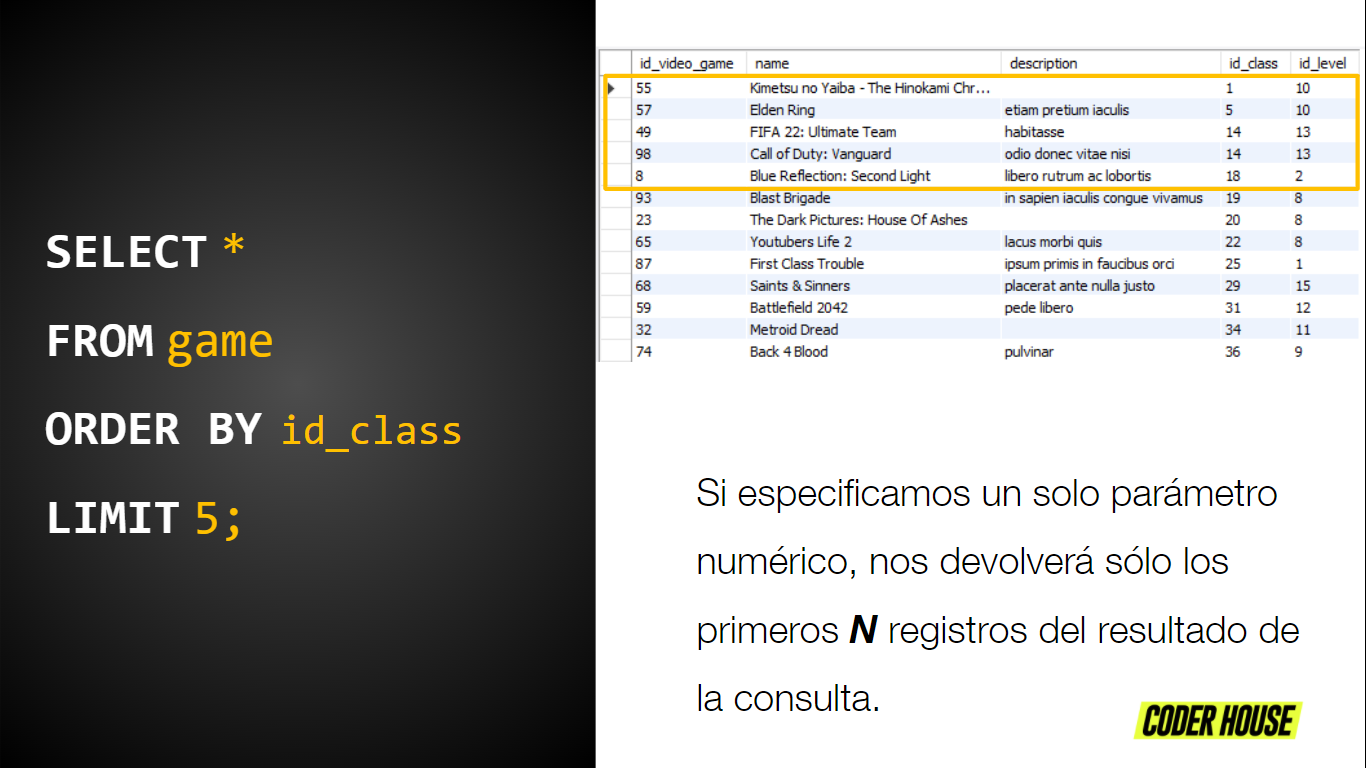
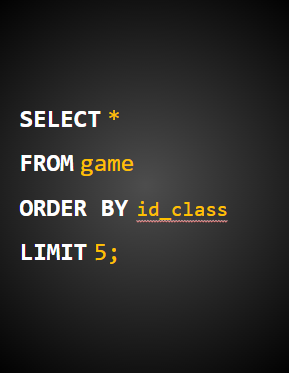
Si no especificamos ninguno, SQL ordena de forma ASC.

* Limit

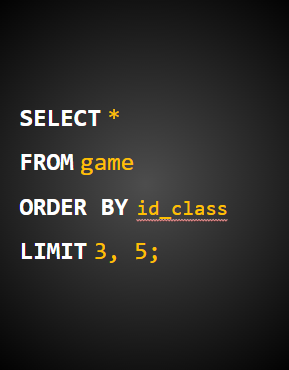
LA SENTENCIA limit

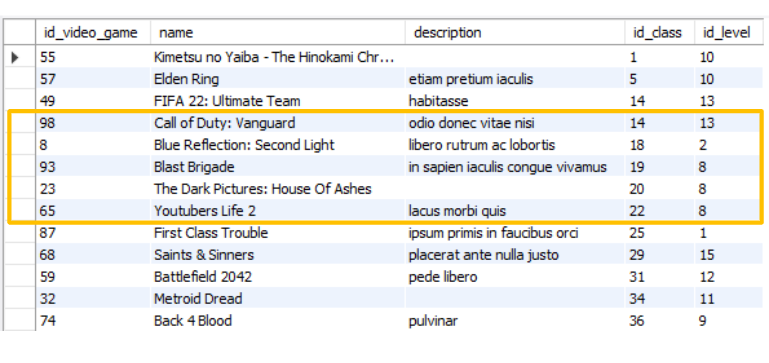
La utilizamos al final de toda la sentencia SELECT, para restringir el número de filas en el resultado de la consulta. LIMIT espera uno o dos parámetros:

* desde qué registro comenzar a mostrar.
* el total de registros próximos a mostrar.



Si especificamos un solo parámetro numérico, nos devolverá sólo los primeros N registros del resultado de la consulta.



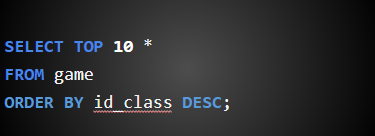


Si especificamos dos parámetros numéricos, se listarán los N registros del resultado de la consulta, partiendo de la posición indicada por el primer parámetro.

USO DE LIMIT TOP

Si alguna vez debes trabajar con SQL Server, ten presente que este motor de BD no incluye la sentencia LIMIT.

En su lugar, utiliza la sentencia TOP para devolverte los primeros N registros de la consulta solicitada. Ejemplo en codigo SQL Server:



ALIAS

¿Cuál es su uso?

SQL Alias es una forma de acotar el nombre de una tabla o columna, simplificando así su uso en sentencias SQL.

Se logra reducir las sentencias SQL cuando incluyen dos o más tablas y/o varios campos.

Se debe usar la palabra reservada AS, seguida del alias que se desea dar a dicho campo o tabla.

La integración de Alias es ideal para acortar el nombre de campos y/o tablas. El uso de Aliases es la forma más fácil de simplificar el nombre de cada campo y/o tabla cuando debemos trabajar con éstos.

En este simple ejemplo no se aprecia o no aporta mucho la simplificación de estos datos, pero cuando comencemos a ver ejemplos más elaborados, donde combinamos dos, tres y hasta cuatro tablas diferentes, con nombres de campos similares, y debamos elaborar agrupaciones, ordenamientos y funciones sobre determinados campos, veremos que utilizar ALIAS será una opción útil, tanto sobre el nombre de un campo como también del nombre de las tablas.

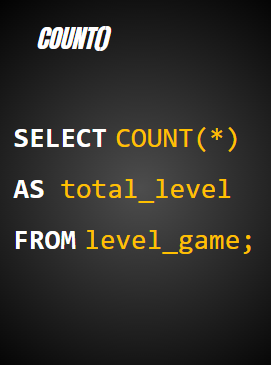
**Funciones de agregación**

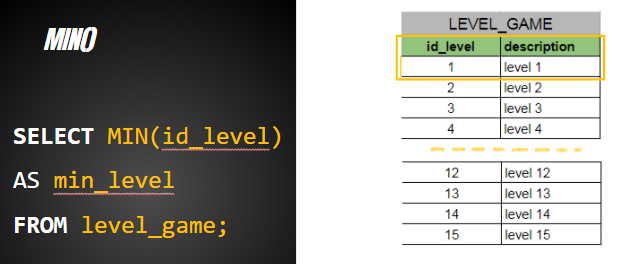
**Definición**

Así como SQL nos permite obtener datos de una o más tablas, también nos permite obtener valores simplificados o resumidos, sobre datos específicos que necesitemos. Esto se conoce como Funciones de Agregación o Agrupación.

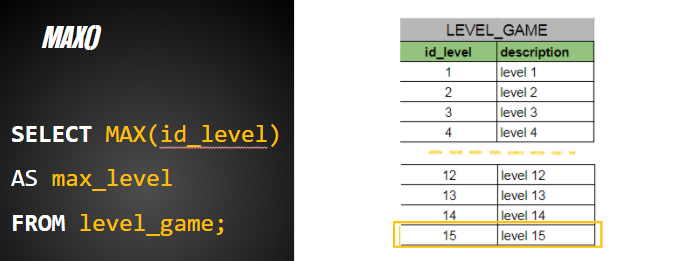
Podemos combinar funciones de totalización, conteo, promedios, valores mínimos y/o máximos, entre otras, al momento de realizar la consulta. Las funciones de agregación se combinan con la cláusula GROUP BY y el uso de AS.

Repaso por las funciones

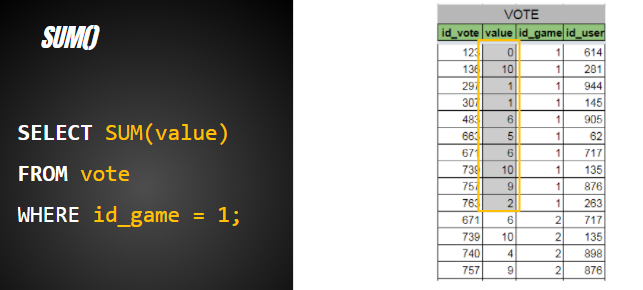
  
Devuelve el número total de filas seleccionadas en una consulta



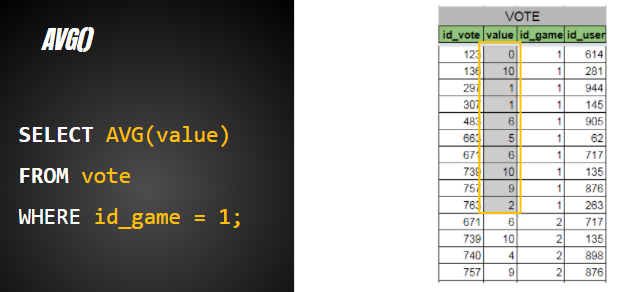
Devuelve el valor mínimo de un campo que especifiquemos



Devuelve el valor máximo de un campo que especifiquemos



Devuelve la suma de los valores de un campo que especifiquemos. La función SUM y AVG, a diferencia de MIN y MAX sólo se utiliza en columnas numéricas. En el ejemplo nos devuelve el puntaje total que ha recibido por los votos el juego cuyo id es 1.



Devuelve el valor promedio de un campo que especifiquemos

La sentencia

SELECT *AVG*(value), *count*(*\**), *sum*(value)

FROM vote

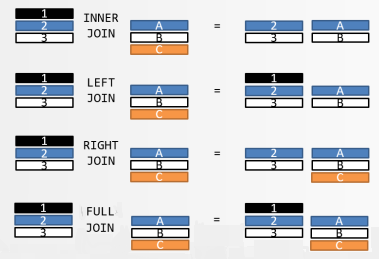
where id\_game = 1;

me devolverá 5, 10 y 50 para los datos indicados en la tabla.

**Join**

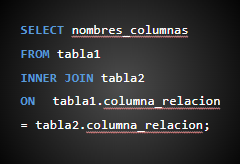
Concepto

JOIN permite combinar registros de diferentes tablas, complementándose con la cláusula ON, la cual establece la condición por la cual queremos unir las tablas. Generalmente son campos comunes entre tablas. Los tipos de JOIN importantes, son cuatro:



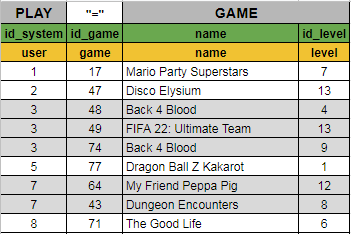
* Inner Join

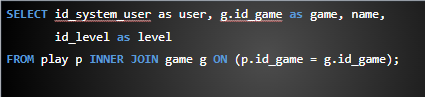
INNER JOIN, o JOIN, retorna todas las filas de las dos tablas siempre que haya coincidencia por el campo declarado en el ON. El resultado es NULL cuando no hay coincidencia alguna.



SINTAXIS: inner join

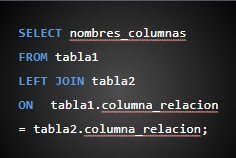
Armamos la consulta utilizando INNER JOIN para llegar al resultado deseado.





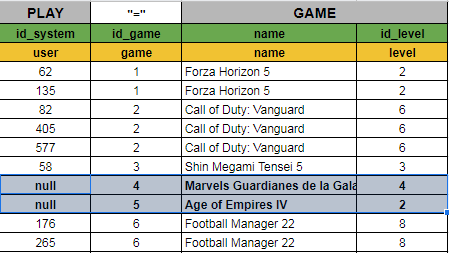
* Left join

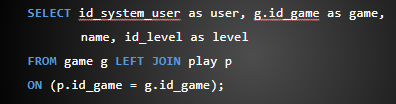
LEFT JOIN, retorna todas las filas de la tabla izquierda que coincidan con las filas de la tabla derecha. El resultado es NULL del lado derecho, cuando no hay coincidencia.



SINTAXIS: left join

Armamos la consulta utilizando LEFT JOIN para llegar al resultado deseado

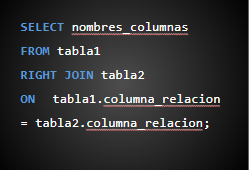




El juego 4 y 5 no es jugado por ningún jugador, “Lista de todos los juegos aún si alguien no los jugó”

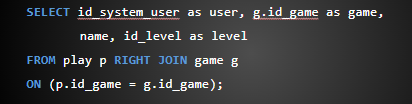
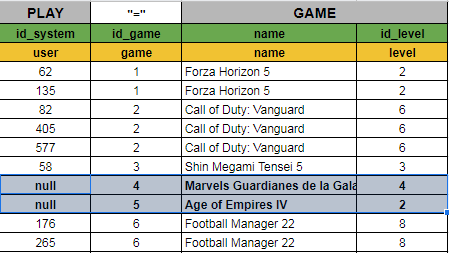
* Right join

RIGHT JOIN, retorna todas las filas de la tabla derecha que coincidan con las filas de la tabla izquierda. El resultado es NULL cuando no hay coincidencia del lado izquierdo.



SINTAXIS: right join

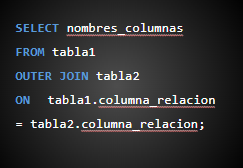
Armamos la consulta utilizando RIGHT JOIN para llegar al resultado deseado



* Full join

FULL JOIN, retorna todas las filas de la tabla derecha y también las filas de la tabla izquierda. Básicamente combina los resultados de LEFT y RIGHT JOIN, pudiendo tener valores nulo de ambos lados..

Nota: MySQL no soporta FULL JOIN.



SINTAXIS: full join

Armamos la consulta utilizando FULL JOIN para llegar al resultado deseado.

