**SQL**

HAY DIVERSAS FORMAS DE BUSCAR DATOS EN NUESTRA BBDD como el método LIKE %%, con este método conseguimos buscar una propiedad que contenga los valores entre los % como por ejemplo en este caso:



Aquí le decimos que nos busque en la tabla **customers las filas que tengan una dirección que contenga el numero 789**, este es el resultado:

Tambien podemos mostrar los **clientes cuyo apellido empiece por W** de la misma forma que antes poniendo solo el % después de la primera letra:



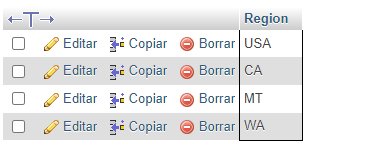
Este sería el resultado:

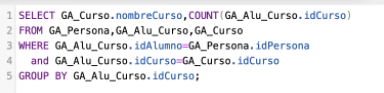


**También podemos ver por ejemplo las regiones que tenemos con clientes.**



Con esto decimos que miremos las regiones en la tabla customers donde la región no sea NULL. Este es el resultado:

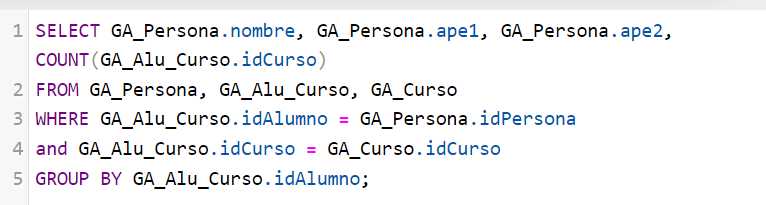




Este código sería para ver los alumnos que hay martriculados en cada curso donde seleccionamos el nombre del curso y contamos los Alu\_Curso con el mismo idCurso que hay en nuestra base de datos y mostrarlo numéricamente de esta forma esto nos dirá cuantos alumnos hay en cada curso:



En cambio, si queremos ver a los alumnos y ver el NÚMERO de cursos en los que están matriculado:



Ahora con este cogemos el nombre y apellidos de las personas y contamos los idCursos que tienen asociados dentro de GA Alu Curso. Aquí veremos a cada alumno y el contador de cursos en los que está matriculado cada alumno:



**INNER JOIN**

**Una JOIN realmente se hace así:**

🡨Así se hace en el SQL en Oracle, por ejemplo.

Esto también se puede hacer con un **INNER** **JOIN**:

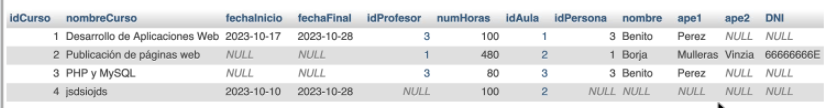


En este caso le decimos que coja la tabla GA\_Curso, y hacemos que Entre en GA\_Persona y que coja los valores donde el id profesor del curso y el id persona coincidan, lo que **viene a ser lo mismo que la query de arriba (JOIN).**

**LEFT OUTER JOIN**



En esta query le decimos que **muestre toda la información de los cursos** pese a que no tenga un profesor asignado. Es decir, **no tiene en cuenta que se cumpla la zona de la derecha** del left outer join.

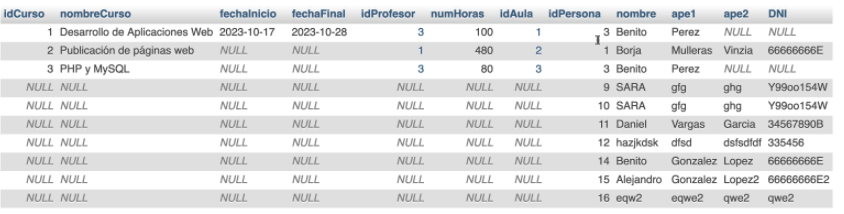


**RIGHT OUTER JOIN**



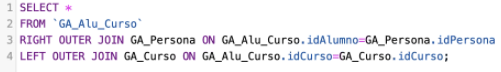
En esta query le decimos que muestre **toda la información** de las personas pese a que no tengan un curso asignado (si tienen curso asignado serian profesores).

Es decir, no tiene en cuenta que se cumpla la zona de la izquierda del right outer join. Hay que tener en cuenta que no salen todas las personas ya que hay alumnos que, si tienen curso asignado, PERO no estamos cogiendo el idAlumno sino el idProfesor.



**Estas funciones de búsqueda se pueden mezclar para mostrar ambas relaciones:**

EN ESTE CASO SI COGEMOS FROM **ALU CURSO**, por lo que tendremos todos los **alumnos** estén en un curso o no lo estén (entonces no serian alumnos sino personas simplemente).



En este caso veremos **todas las personas (no profesores)** que tenemos **estén asociadas (alumnos) o no a un curso (personas).**



Con esta función AVG en la query haremos la media de la columna que tengamos entre paréntesis de la tabla productos en este caso



La media de stock es de 25.

También podemos consultar los productos que tengamos con un stock inferior a la media:



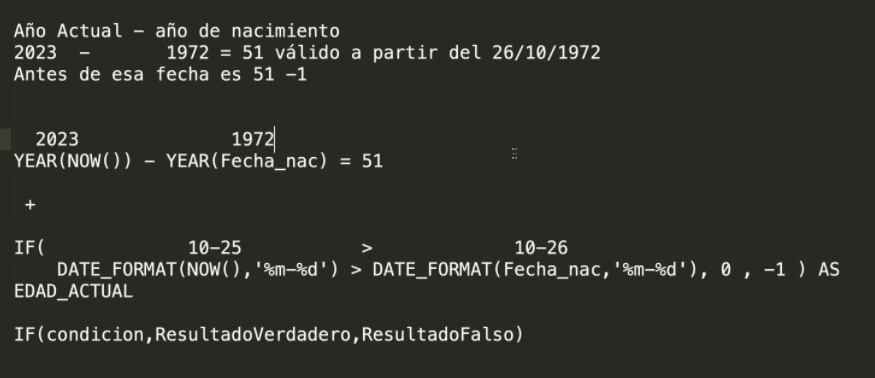
  
En este caso en el que la media es 25, nos seleccionará los que su Stock sea inferior a esa media.

Aquí con esta query por ejemplo seleccionaremos de la tabla precio en la columna precio el producto con el precio máximo (es decir, el más caro).

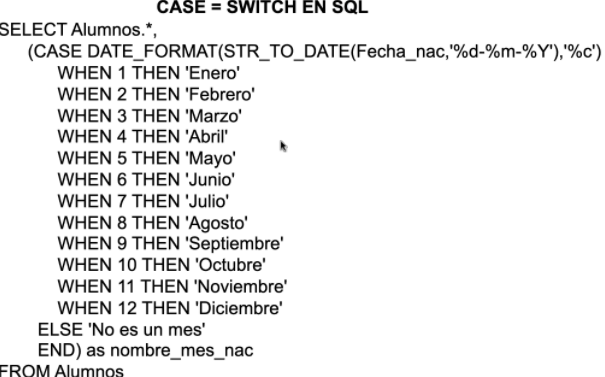


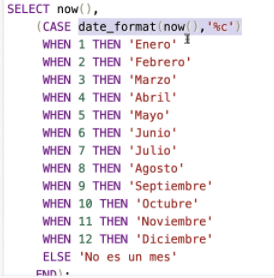
Con el date\_format() podemos coger una fecha para que aparezca en formato distinto. En el siguiente caso pondremos que la fecha nos aparezca en dia-mes-año



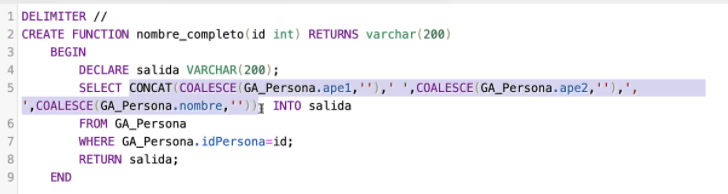
Esquema de como calcular la fecha de nacimiento.

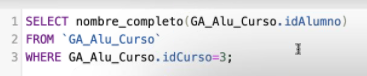
**SWITCH EN SQL:**



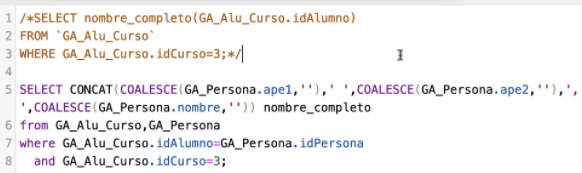


El % C es para coger 2 numeros del mes en caso de que tenga 2 cifras.



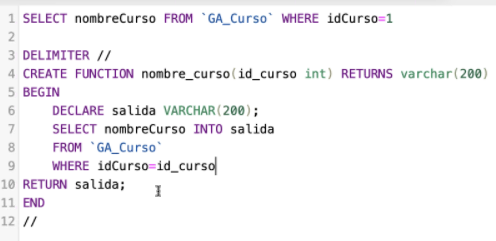


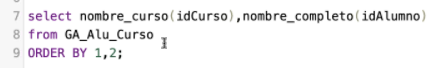
Esta función dará el nombre de los alumnos que están en el idCurso 3

Hacer esa codificación es exactamente lo mismo que hacer esto:  


En temas de producción, es más rápido hacer una query completa que la función ya que además del llamado a la función, también

estamos ejecutando esa misma query prácticamente dentro de la función.

tambien podemos crear la funcion nombreCurso



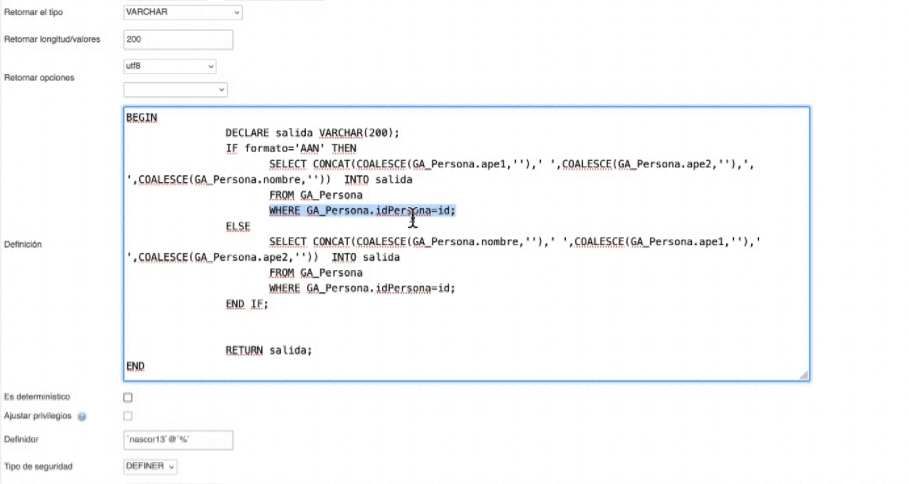
y mezclar ambas funciones para ver el nombre del alumno y el curso en el que está.

EXPRESION NVL2(expresion1, expresion2, expresion3)



La expresion NVL2 determina un valor null, si el primer parámetro es null, nos devolverá el parámetro 3, sino nos devolverá el parámetro 2.

En este caso si la fecha de inicio es null nos devolverá la palabra NULO y sino pues nos devolverá la función formato\_mes\_esp.

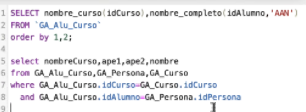


Esta función nos permitirá introducir el parámetro de la function si queremos mostrar el nombre en formato AAN o NAA.

Aquí llamamos 2 veces a la función una sin especificar el 2º parámetro y otro con AAN, podremos ver los dos formatos en la salida donde estarán los dos formatos, NAA y AAN.



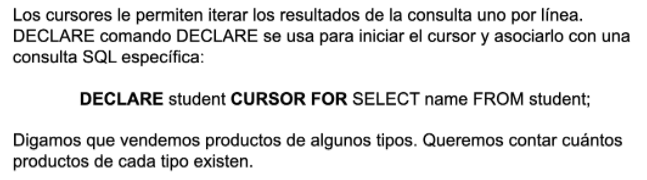
Con esto en GA\_Alu\_Curso podremos ver el nombre de cada alumno que hay en cada curso.

El llamado a la función línea 1-3 es equivalente a ls codificación de debajo con select, from y where donde es ‘’más codigo’’ pero es bastante más eficiente.



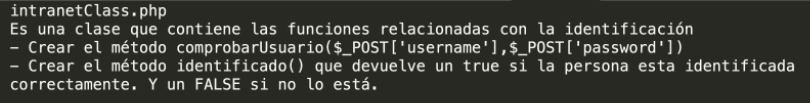
Usar la Query de select tarda 0,0003s mientras usar la función tarda 0,0007s que es el doble. En este caso el tiempo es bajo en ambos caso y no notaríamos la diferencia pero tal vez debamos tener cuidado cuando la cantidad de datos en la BBDD llega a cierta cantidad.

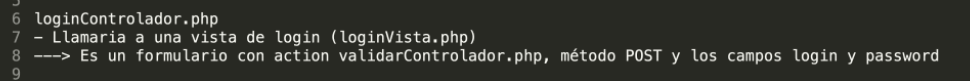
**CURSORES:**



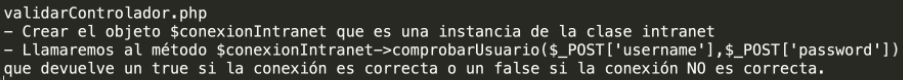
**EJEMPLO DE CURSOR**

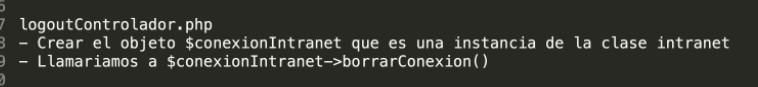
DELIMITER //  
CREATE DEFINER=`nascor13`@`%` PROCEDURE `product\_count`()  
BEGIN  
 DECLARE p\_type VARCHAR(255);  
 DECLARE p\_count INT(10) UNSIGNED;  
 DECLARE done INT DEFAULT 0;  
 DECLARE product CURSOR FOR  
   SELECT type, COUNT(\*)  
   FROM CURSOR\_product  
   GROUP BY type;  
 DECLARE CONTINUE HANDLER FOR SQLSTATE '02000' SET done = 1;  
 TRUNCATE CURSOR\_product\_type\_count;  
   
    OPEN product;  
 REPEAT FETCH product INTO p\_type, p\_count;  
  IF NOT done THEN  
   INSERT INTO CURSOR\_product\_type\_count  
   SET type = p\_type, count = p\_count;  
  END IF;  
 UNTIL done   
    END REPEAT;  
 CLOSE product;  
END//  
DELIMITER ;

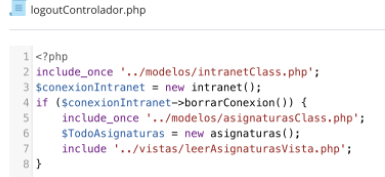














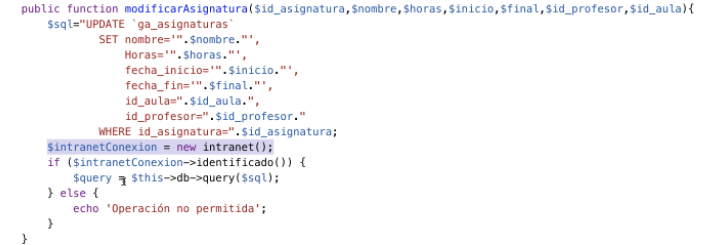


$conexionIntranet = new intranet();

If( $conexionIntranet->identificado()){

…

}else {…}



Cuando se hagan operaciones para las que queremos que esten identificados, usaremos la función dentro de la función.



COMPROBAR EL NIVEL DE ACCESO POR EJEMPLO PARA COMPROBAR SI EL USUARIO ES SIMPLEMENTE USER O ES ADMIIN (QUE TENDRÁ PRIVILEGIOS EN LA PAGINA COMO AÑADIR LIBROS, MODIFICAR…)

