

200 - Funciones de fila (Hecho, DAM) Juan Francisco Sanchez

201 Escriba una consulta para mostrar la fecha actual. Etiquete la columna Date.

- `SELECT NOW() AS "Date";`

202 Para cada empleado, muestre el número de empleado, el apellido, el salario y el sueldo con un aumento del 15% y expresado como número entero redondeado, etiquétala como "Nuevo sueldo".

- `SELECT empleado_id, apellido, sueldo, ROUND(sueldo + sueldo * 0.15) AS "Nuevo Sueldo"`

`FROM empleados;`

203 Modifique la consulta anterior para agregar una columna que reste el salario nuevo al antiguo . Etiquete la columna como "Incremento".

- `SELECT empleado_id, apellido, sueldo,`
- `ROUND(sueldo + sueldo * 0.15) AS "Nuevo Sueldo",`
- `ROUND(sueldo + sueldo * 0.15) - sueldo AS "Incremento"`
- `FROM empleados;`

204 "Escriba una consulta que muestre el apellido del empleado con la primera letra en mayúsculas y todas las demás en minúsculas así como la longitud del apellido de todos los empleados cuyos apellidos comiencen por J, A o M. Ordene los resultados por apellidos de empleado. Etiquete la columna como "Apellido" y "Longitud".

- `SELECT apellido AS "Apellido", LENGTH(apellido) AS "Longitud"`
- `FROM empleados`
- `WHERE SUBSTR(UPPER(apellido),1,1) IN ('J', 'A', 'M')`
- `ORDER BY apellido;`

205 Para cada empleado, muestre su apellido y calcule el número de meses entre el día de hoy y la fecha de contratación. Etiquete la columna como "Meses trabajados", ordene los resultados según el número de meses trabajados (de menos a más). Redondee el número de meses al número entero más próximo.

- `SELECT apellido, ROUND(TIMESTAMPDIFF(MONTH, fecha_contrato, NOW())) AS "Meses trabajados"`
- `FROM empleados`
- `ORDER BY "Meses trabajados";`

206 Cree una consulta para mostrar el apellido y el sueldo de todos los empleados. Formatee el sueldo para que tenga 15 caracteres de longitud, rellenado a la izquierda con €. Etiquete la columna como "Sueldo".

- `SELECT apellido, LPAD(sueldo, 15, '€') AS "Sueldo"`
- `FROM empleados;`

207 Escriba una consulta que produzca lo siguiente para cada empleado: <Apellido> gana <salario>€ pero le gustaría ganar <salario*3>€. Llame a la columna Salario soñado. (Ej: "Foster gana 3300€ pero le gustaría ganar 9900€.")

- SELECT CONCAT(apellido, ' gana ', sueldo, '€ pero le gustaría ganar ', sueldo * 3, '€.') AS `Salario soñado`
- FROM empleados;

208 Muestre el apellido, la fecha de contratación y el día de la semana en el que comenzó el empleado a trabajar (LUNES, MARTES,...)(Mayúsculas). Etiquete la columna como "Día". Ordene los resultados por día de la semana, comenzando por el lunes

- SELECT apellido, fecha_contrato, DAYNAME(fecha_contrato) AS 'Día'
- FROM empleados
- ORDER BY DAYOFWEEK(fecha_contrato);

209 Muestre el apellido de cada empleado, así como la primera fecha de revisión de salario, que es el primer lunes después de cada seis meses empezando a contar desde la fecha de contratación. Etiquete la columna "Revisión". Formatea la fecha para que aparezca en un formato similar a "Lunes 10 de Octubre de 2016".

- SELECT apellido, DATE_FORMAT(DATE_ADD(DATE_ADD(fecha_contrato, INTERVAL 6 MONTH),
- INTERVAL (9 - DAYOFWEEK(DATE_ADD(fecha_contrato, INTERVAL 6 MONTH))) % 7 DAY), '%W %d de %M de %Y') AS `Revisión`
- FROM empleados;

210 Cree una consulta que muestre el identificador del pedido, el estado y identificador del vendedor. Si un pedido no tiene vendedor, ponga "No vendedor". Etiquete la columna como "Vendedor".

- SELECT pedido_id, estado, IFNULL(CAST(vendedor_id AS CHAR), 'No vendedor') AS "Vendedor"
- FROM pedidos;

211 "Cree una consulta que muestre el apellido de los empleados y que indique las cantidades de sus salarios anuales con asteriscos en una sola columna. Cada asterisco significa mil dólares. Ordenado por el sueldo de mayor a menor. Etiquete la columna como "sueldos".

- SELECT CONCAT(apellido, ' ', REPEAT('*', FLOOR(sueldo / 1000))) AS `sueldos`
- FROM empleados
- ORDER BY sueldo DESC;

212 "Utilizando la función CASE, escriba una consulta que muestre el apellido, puesto de trabajo y el grado de todos los empleados basándose en el valor del puesto de trabajo, según los siguientes datos (etiqueta la columna como "Grado"):

- SELECT apellido, puesto_trabajo,
- CASE puesto_trabajo
- WHEN 'President' THEN 'A'
- WHEN 'Administration Vice President' THEN 'B'
- WHEN 'Stock Manager' THEN 'C'
- WHEN 'Sales Representative' THEN 'D'
- WHEN 'Shipping Clerk' THEN 'E'
- ELSE '0'
- END
- AS "Grado"
- FROM empleados;

213 Vuelva a escribir la sentencia de la pregunta anterior utilizando el DECODE.

- SELECT apellido, puesto_trabajo,
- CASE puesto_trabajo
- WHEN 'President' THEN 'A'
- WHEN 'Administration Vice President' THEN 'B'
- WHEN 'Stock Manager' THEN 'C'
- WHEN 'Sales Representative' THEN 'D'
- WHEN 'Shipping Clerk' THEN 'E'
- ELSE '0'
- END
- AS "Grado"
- FROM empleados;

214 Cree una consulta que muestre el contacto_id, apellido y teléfono de aquellos contactos los cuales el segundo conjunto de números del teléfono sea 49 o bien estar entre el 70 y 90. Ordenar el resultado en orden ascendente por el identificador del contacto (Dedicado a Alejandro y Nacho).

- SELECT contacto_id, apellido, telefono
- FROM contactos
- WHERE LENGTH(RTRIM(SUBSTR(telefono, 5, 3))) = 2 AND
- SUBSTR(telefono, 5, 2) BETWEEN 70 AND 90 OR
- SUBSTR(telefono, 5, 2) = 49
- ORDER BY contacto_id;