1. - Funciones de fila (Hecho, DAM) Juan Francisco Sanchez
2. Escriba una consulta para mostrar la fecha actual. Etiquete la columna Date.
   * SELECT NOW() AS "Date";
3. 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;

1. 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;
2. "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;
3. Para cada empleado, muestre su apellido y calcule el número de meses entre el dia 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";
4. 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;
5. Escriba una consulta que produzca lo siguiente para cada empleado: <ApeIIido> gana <saIario>€ pero le gustaría ganar <saIario“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;
6. 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 dia de la semana, comenzando por el lunes
   * SELECT apellido, fecha\_contrato, DAYNAME(fecha\_contrato) AS 'Día'
   * FROM empleados
   * ORDER BY DAYOFWEEK(fecha\_contrato);
7. 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 las 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;
8. 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;
9. "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. Etiqueta la columna como "sueldos".
   * SELECT CONCAT(apellido, ' ', REPEAT('\*', FLOOR(sueldo / 1000))) AS `sueldos`
   * FROM empleados
   * ORDER BY sueldo DESC;

1. "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;
2. 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;
3. 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;