

Jesús Manuel Juárez Posillas.

26/10/21

1. Consulta el apellido de los empleados donde su apellido tiene una letra "o" en su segunda posición y lo demás puede tener lo que sea.
2. Consulta el employee_id, last_name, job_id de los empleados donde el job_id contenga un "SA_\" donde sea en la cadena.
3. Consulta el last_name, job_id, salary de los empleados donde el job_id sea igual a "SA_REP" o igual a "AD_PRES" y tengan un salario mayor a 15000.
4. Consulta el last_name, job_id, department_id y el hire_date de los empleados, y lo ordena ~~en~~ usando la posición numérica de las columnas.
5. Consulta el employee_id, last_name, job_id y pide el nombre de otra columna a consultar de los empleados donde pide una condición la cual debe cumplirse, además de ordenar el resultado por la columna que se le indique.

6. Consulta el empleado id, además de concatenar el nombre y apellido colocándole al resultado el nombre 'name' (nombre de columna), también consulta el job_id, obtiene la longitud del apellido, además de obtener la posición de la letra 'a' (si es que tiene), de lo contrario es 0, llamando a la columna "Contains 'a'" de los empleados donde la subcadena de job_id de la posición 4 al final de la cadena sea igual a "REP".

```
7. Select concat(concat(concat('The job_id  
for', first_name), 'is '), job_id  
lower(job_id)) from Employees  
"EMPLOYEE DETAILS" from Employees;
```

8. Consulta el resultado de truncar 45.923 a 2 decimales que quedaría 45.92, también trunca el mismo número pero sin especificar decimales, por lo que quedaría 45 y por último trunca el mismo número pero con un -1 lo que indica que serán números enteros los que se afecten lo cual resulta en 40.

9. RR

1995

2017

2017

1995

YY

1995

1917

2017

2095

10. Select employee_id, To_char(hire_date, 'mm/yy') ~~from Emp~~ Month_Hired
from Employees where employee_id = 205;

11. Select last_name, to_char(hire_date, 'fm parrth"of" Month yyyy hh24:mi:ssAm')
Hire_date from Employees;

12. Para quitar espacios en blanco inutilizados,
✶ quitar ceros a la izquierda.

13. Especifica la coincidencia exacta para
el argumento del caracter.

14. NVL: Verifica si el primer argumento es null, si no lo es regresa el primer argumento si lo es regresa el segundo.

NVL2: Verifica que el primer argumento no es null, si no lo es regresa el segundo arg. Pero si lo es regresa el tercer arg.

NULLIF: Verifica que los dos argumentos sean iguales, si lo son regresa el primero si no, regresa null.

COALESCE: Regresa el primer argumento que no sea null.

15. Select last_name, salary, commission_pct,
nvl2(commission_pct, salary + ((salary/100) *
(commission_pct * 100)), salary + 2000) ~~new_salary~~
"New Salary" from Employees;

16. Select last_name, salary,
decode(trunc(salary/2000, 0),
0, 0.00,
1, 0.09,
2, 0.20,
3, 0.30,
4, 0.40,
5, 0.42, 6, 0.44, 0.45)
tax_rate from Employees;

17: select department_id, avg(salary)
from Employees group by department_id

18: consulta los department_id y el promedio de los salarios agrupandolos por el department_id, y solo va a mostrar los resultados de los promedios que sean mayores a 10000.

19: select to_char(sysdate, 'fm"Hor es: "
day dd "de" Month "de" yyyy "' y son las: "
hh24:mi:ss Am') from dual;

20: select department_id, ~~count(*)~~ count(*)
from Employees where commission_pct is
not null group by department_id
having count(*) > 5;