

Práctica SQL 12c

1. Otras Subconsultas

• Seleccionar el nombre, salario y departamento de los empleados que ganen mas que cualquiera de los salarios máximos de los departamentos 50, 60 y 70. Usar el operador ANY

♦ FIRST_NAME		DEPARTMENT_ID
Steven	24000	90
Neena	17000	90
Lex	17000	90
Nancy	12008	100
Den	11000	30
John	14000	80
Karen	13500	80
Alberto	12000	80
Gerald	11000	80
Eleni	10500	80
Peter	10000	80
David	9500	80
Janette	10000	80
Patrick	9500	80
Clara	10500	80
Danielle	9500	80

 Indicar el nombre de los departamentos cuyo salario medio sea superior a 9000. Usar el operador IN

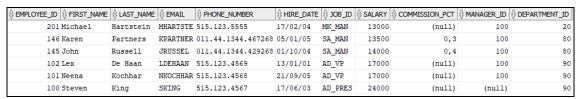


• Indicar el nombre del empleado, el nombre del departamento, el salario de los empleados que tengan el salario máximo de su departamento. Ordenado por salario descendentemente. Usar el operador IN



FIRST_NAME		
Steven	Executive	24000
John	Sales	14000
Michael	Marketing	13000
Nancy	Finance	12008
Shelley	Accounting	12008
Den	Purchasing	11000
Hermann	Public Relations	10000
Alexander	IT	9000
Adam	Shipping	8200
Susan	Human Resources	6500
Jennifer	Administration	4400

- Realizar la misma consulta anterior pero usando una subconsulta sincronizada
- Indicar los datos de los empleados que ganen más que todos los empleados del departamento 100. Usar el operador ALL

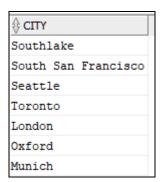


Mostrar los empleados que tienen el mayor salario de su departamento.
 Usar subconsultas sincronizadas.

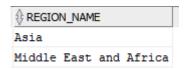
♦ DEPARTMENT_ID		
100	Nancy	12008
30	Den	11000
90	Steven	24000
20	Michael	13000
70	Hermann	10000
110	Shelley	12008
50	Adam	8200
80	John	14000
40	Susan	6500
60	Alexander	9000
10	Jennifer	4400

 Visualizar las ciudades en las que haya algún departamento. Debemos usar consultas sincronizadas y el operador EXISTS





• Visualizar el nombre de las regiones donde no hay departamentos. Usar subconsultas sincronizadas y el operador NOT EXISTS





SOLUCIONES

 Seleccionar el nombre, salario y departamento de los empleados que ganen más que cualquiera de los salarios máximos de los departamentos 50, 60 y 70. Usar el operador ANY

select first_name, salary, department_id from employees where SALARY > ANY (select Max(salary) from employees group by department_id HAVING department_id in ('50','60','70'));

 Indicar el nombre de los departamentos cuyo salario medio sea superior a 9000. Usar el operador IN

SELECT DEPARTMENT NAME FROM DEPARTMENTS

WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID FROM EMPLOYEES

GROUP BY DEPARTMENT_ID

HAVING AVG(SALARY)> 9000);

 Indicar el nombre del empleado, el nombre del departamento, el salario de los empleados que tengan el salario máximo de su departamento. Ordenado por salario descendentemente. Usar el operador IN

SELECT FIRST_NAME, DEPARTMENT_NAME, SALARY

FROM EMPLOYEES JOIN DEPARTMENTS USING (DEPARTMENT_ID)

WHERE (DEPARTMENT_ID,SALARY)

IN(SELECT

DEPARTMENT_ID,MAX(SALARY)

FROM EMPLOYEES

GROUP BY DEPARTMENT_ID)

ORDER BY SALARY DESC;

Realizar la misma consulta anterior pero usando una subconsulta sincronizada

SELECT FIRST_NAME, DEPARTMENT_NAME, SALARY

FROM EMPLOYEES EMPLE JOIN DEPARTMENTS DEPT ON (EMPLE.DEPARTMENT_ID=DEPT.DEPARTMENT_ID)

WHERE SALARY =(SELECT MAX(SALARY)

FROM EMPLOYEES

WHERE DEPARTMENT_ID=EMPLE.DEPARTMENT_ID

GROUP BY DEPARTMENT_ID)

ORDER BY SALARY DESC;

 Indicar los datos de los empleados que ganen más que todos los empleados del departamento 100. Usar el operador ALL

SELECT * FROM EMPLOYEES

WHERE SALARY > ALL (SELECT SALARY FROM EMPLOYEES WHERE DEPARTMENT_ID=100);

Mostrar los empleados que tienen el mayor salario de su departamento.
 Usar subconsultas sincronizadas.



SELECT DEPARTMENT_ID,FIRST_NAME, SALARY FROM EMPLOYEES EMPLE WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEES WHERE DEPARTMENT_ID = EMPLE.DEPARTMENT_ID);

 Visualizar las ciudades en las que haya algún departamento. Debemos usar consultas sincronizadas y el operador EXISTS

SELECT CITY FROM LOCATIONS LOCALIDADES

WHERE EXISTS (SELECT * FROM DEPARTMENTS WHERE LOCATION_ID=LOCALIDADES.LOCATION_ID);

 Visualizar el nombre de las regiones donde no hay departamentos. Usar subconsultas sincronizadas y el operador NOT EXISTS

SELECT REGION_NAME FROM REGIONS REGIONES

WHERE NOT EXISTS (SELECT * FROM COUNTRIES NATURAL JOIN LOCATIONS NATURAL JOIN DEPARTMENTS

WHERE REGION_ID=REGIONES.REGION_ID);

•