

Práctica

18. Subconsultas

- 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.

```
-- 1)
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'));
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 31 en 0,012 segundos

	FIRST_NAME	SALARY	DEPARTMENT_ID
1	Steven	24000	90
2	Neena	17000	90
3	Lex	17000	90
4	John	14000	80
5	Karen	13500	80
6	Michael	13000	20
7	Shelley	12008	110
8	Nancy	12008	100
9	Alberto	12000	80
10	Lisa	11500	80
11	Gerald	11000	80

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

```
-- 2)
SELECT department_name
FROM departments
WHERE (department_id) IN (SELECT DEPARTMENT_ID
FROM employees
GROUP BY department_id
HAVING ROUND(AVG(salary)) > 9000);

-- 3)
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 4 en 0,014 segundos

	DEPARTMENT_NAME
1	Marketing
2	Public Relations
3	Executive
4	Accounting

- 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 descendientemente. Usar el operador IN

```

SELECT e.first_name, e.salary, d.department_name
FROM employees e
JOIN departments d ON e.department_id = d.department_id
WHERE (e.department_id, e.salary) IN (SELECT department_id, MAX(salary)
                                     FROM employees
                                     GROUP BY department_id)
ORDER BY e.salary DESC;

```

Resultado de la Consulta x

Todas las Filas Recuperadas: 11 en 0,07 segundos

	FIRST_NAME	SALARY	DEPARTMENT_NAME
1	Steven	24000	Executive
2	John	14000	Sales
3	Michael	13000	Marketing
4	Nancy	12008	Finance
5	Shelley	12008	Accounting
6	Den	11000	Purchasing
7	Hermann	10000	Public Relations
8	Alexander	9000	IT
9	Adam	8200	Shipping
10	Susan	6500	Human Resources
11	Jennifer	4400	Administration

- Realizar la misma consulta anterior pero usando una subconsulta sincronizada

```

-- 4)
SELECT emp.first_name, d.department_name, emp.salary
FROM employees emp
JOIN departments d ON emp.department_id = d.department_id
WHERE salary = (SELECT MAX(salary) FROM employees
               WHERE department_id = emp.department_id)
ORDER BY salary DESC;

```

Resultado de la Consulta x

Todas las Filas Recuperadas: 11 en 0,03 segundos

	FIRST_NAME	DEPARTMENT_NAME	SALARY
1	Steven	Executive	24000
2	John	Sales	14000
3	Michael	Marketing	13000
4	Nancy	Finance	12008
5	Shelley	Accounting	12008
6	Den	Purchasing	11000
7	Hermann	Public Relations	10000
8	Alexander	IT	9000
9	Adam	Shipping	8200
10	Susan	Human Resources	6500
11	Jennifer	Administration	4400

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

```
-- 5)
SELECT *
FROM employees
WHERE salary > ALL (SELECT MAX(SALARY)
FROM employees
GROUP BY DEPARTMENT_ID
HAVING department_id IN('100'));
```

resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 6 en 0,013 segundos

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
201	Michael	Hartstein	MHARTSTE	515.123.5555	17/02/04	MK_MAN	13000	(null)	100	100
146	Karen	Partners	KPARTNER	011.44.1344.467268	05/01/05	SA_MAN	13500	0,3	100	100
145	John	Russell	JRUSSEL	011.44.1344.429268	01/10/04	SA_MAN	14000	0,4	100	100
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	17000	(null)	100	100
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	100
100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	100

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

```
-- 6)
SELECT department_id, first_name, salary
FROM employees emp
WHERE salary = (SELECT MAX(salary) FROM employees
WHERE department_id = emp.department_id);
```

Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 11 en 0,032 segundos

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

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

```
-- 7)
SELECT city
FROM locations loc
WHERE EXISTS (SELECT * FROM departments WHERE location_id = loc.location_id);
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 7 en 0,002 segundos

	CITY
1	Southlake
2	South San Francisco
3	Seattle
4	Toronto
5	London
6	Oxford
7	Munich

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

```
-- 9)
SELECT region_name FROM REGIONS Regiones
WHERE NOT EXISTS (SELECT * FROM countries NATURAL JOIN locations NATURAL JOIN departments
WHERE region_id=Regiones.region_id);
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,01 segundos

	REGION_NAME
1	Asia
2	Middle East and Africa