

LISTA 7 - RESPOSTAS

SUBCONSULTAS, OPERADORES PARA TESTE DE CONJUNTO DE VALORES

1 - Utilize uma subconsulta na cláusula SELECT a fim de mostrar o nome completo do funcionário, sua função, o salário máximo referente a sua função e o salário atual. Ordene o relatório pelo nome.

Nome Completo	Função	Salário máximo da função	Salário Atual
Alcemar Illuminati	President	20000.00	23300.00
Carlito Silva	Seller	5000.00	800.00
Denis Snider	Counter	7000.00	6584.66
Douglas Müller	Seller	5000.00	2000.00
Emilio Hennemann	Administrator	5000.00	2800.00
Joana Souza	Analyst	8000.00	5000.00
João Kipper	Counter	7000.00	4444.05
João Menezes	Administrator	5000.00	1500.00
João Silva	Seller	5000.00	800.00
Lucas Camargo	Engineer	10000.00	8900.55
Lucas Lima	Administrator	5000.00	1010.00
Marcolino Souza	Administrator	5000.00	1840.00
Marcos Santos	Seller	5000.00	1200.00
Maria Antonieta	Support	5000.00	3000.00
Maurício Batista	Programmer	6000.00	3500.00
Paulo Campos	Support	5000.00	2500.00
Pedro Rocha	Director	15000.00	16500.00
Sofia Horn	Counter	7000.00	2000.00
Valesca Mattos	Secretary	5000.00	6000.00
Vanessa Hunemeier	Engineer	10000.00	7500.00

```
SELECT e.employee_first_name || ' ' || e.employee_last_name AS "Nome Completo",
j.job_name AS "Função", (
    SELECT job_max_salary
    FROM job j
    WHERE j.job_id = e.job_id) AS "Salário máximo da função", e.employee_salary AS
"Salário Atual"
FROM employee e INNER JOIN job j ON e.job_id = j.job_id
ORDER BY 1;
```

2 - Utilize uma subconsulta na cláusula SELECT a fim de mostrar o primeiro nome do funcionário e o nome do seu departamento.

Nome	Departamento
Carlito	Sales
João	Sales
Marcos	Sales
Paulo	Holder
Maurício	TI
João	Financial
Sofia	Financial
Lucas	Engineering
Marcolino	Human Resources
Lucas	Human Resources
João	Human Resources
Douglas	Sales
Maria	Holder
Joana	TI
Denis	Financial
Vanessa	Engineering
Emilio	Human Resources
Valesca	Direction
Pedro	Direction
Alcemar	Direction

```
SELECT e1.employee_first_name AS "Nome", (  
    SELECT d2.department_name  
    FROM department d2  
    WHERE e1.department_id = d2.department_id  
    ) AS "Departamento"  
FROM employee e1;
```

3 - Utilize uma subconsulta na cláusula SELECT a fim de mostrar o nome de todos os funcionários, com seus respectivos salários e o percentual do salário em relação a renda de toda a empresa. O relatório deverá ser organizado do menor ao maior salário.

Nome	Salário	Percentual do Total %
João	800.00	0.79
Carlito	800.00	0.79
Lucas	1010.00	1.00
Marcos	1200.00	1.19
João	1500.00	1.48
Marcolino	1840.00	1.82
Sofia	2000.00	1.98
Douglas	2000.00	1.98
Paulo	2500.00	2.47
Emilio	2800.00	2.77

Maria	3000.00	2.97
Maurício	3500.00	3.46
João	4444.05	4.39
Joana	5000.00	4.94
Valesca	6000.00	5.93
Denis	6584.66	6.51
Vanessa	7500.00	7.41
Lucas	8900.55	8.80
Pedro	16500.00	16.31
Alcemar	23300.00	23.03

```

SELECT e.employee_first_name as "Nome", e.employee_salary as "Salário", ROUND ( (
    SELECT (e.employee_salary * 100) / SUM (e1.employee_salary)
    FROM employee e1) :: numeric,2) AS "Percentual do Total %"
FROM employee e
ORDER BY e.employee_salary;

```

4 - Utilize uma subconsulta na cláusula SELECT a fim de mostrar o nome completo, o salário, o nome do cargo, o salário mínimo e máximo, referente ao cargo dos empregados que são vendedores e possuam salários superior a R\$1.500,00.

Nome	Salário	Cargo	Salário Mín	Salário Max
Douglas Müller	2000.00	Seller	500.00	5000.00

```

SELECT employee_first_name || ' ' || employee_last_name AS "Nome", employee_salary AS
"Salário", (
    SELECT job_name
    FROM job
    WHERE job_id = 1) AS "Cargo"
, (
    SELECT job_min_salary
    FROM job
    WHERE job_id = 1) AS "Salário Mín"
, (
    SELECT job_max_salary
    FROM job
    WHERE job_id = 1) AS "Salário Max"
FROM employee
WHERE job_id = 1 AND employee_salary > 1500;

```

5 - Utilize uma subconsulta na cláusula FROM a fim de listar o código, nome, salário e nome do departamento de todos os empregados que tem código maior de 9 e recebem mais de R\$2.000,00. Organizar o relatório com os códigos de em ordem crescente.

Código	Nome	Salário	Departamento
10	João	4444.05	Finacial
12	Vanessa	7500.00	Engineering
13	Lucas	8900.55	Engineering
14	Emilio	2800.00	Human Resources
18	Valesca	6000.00	Direction
19	Pedro	16500.00	Direction
20	Alcemar	23300.00	Direction

```
SELECT DISTINCT e.employee_id AS "Código", e.employee_first_name AS "Nome",  
e.employee_salary AS "Salário", d.department_name AS "Departamento"  
  
FROM department d, (  
  
    SELECT DISTINCT employee_id, employee_first_name, employee_salary, department_id  
  
    FROM employee  
  
    WHERE employee_id > 9  
  
    AND employee_salary > 2000) e  
  
WHERE e.department_id = d.department_id  
  
ORDER BY e.employee_id;
```

6 - Utilize uma subconsulta na cláusula FROM a fim de mostrar o nome, sobrenome, e seu respectivo nome de departamento, contudo, apenas dos funcionários que foram admitidos a partir de 2012 devem aparecer na listagem. Ordene o relatório pelo primeiro nome.

Nome	Sobrenome	Departamento
Carlito	Silva	Sales
João	Menezes	Human Resources
João	Kipper	Finacial
Marcos	Santos	Sales
Paulo	Campos	Holder
Sofia	Horn	Finacial
Valesca	Mattos	Direction

```
SELECT e.employee_first_name AS "Nome", e.employee_last_name AS "Sobrenome",  
d.department_name AS "Departamento"  
  
FROM department d, (  
  
    SELECT e.employee_id, e.employee_first_name, e.employee_last_name,  
    e.department_id  
  
    FROM employee e  
  
    WHERE DATE_PART ('year', employee_hire) >= 2012) e  
  
WHERE d.department_id = e.department_id  
  
ORDER BY 1;
```

7 - Utilize uma subconsulta na cláusula FROM a fim de mostrar todos os departamentos cujo salário de seu gerente seja maior que 2000.

Departamentos
Direction
Engineering
Financial
Holder
Human Resources
TI

```
SELECT department.department_name AS "Departamentos"
FROM (
    SELECT employee_id
    FROM employee
    WHERE employee_salary > 2000) gerentes
INNER JOIN department ON gerentes.employee_id = department.manager_id
ORDER BY 1;
```

8 - Utilize uma subconsulta na cláusula FROM a fim de mostrar o nome de todos os departamentos, cuja localização esteja vinculada a uma região. Mostre também o nome do país e da região brasileira, ordenando pelo nome da cidade.

department_name	location_name	region_name
Purchasing	Franca	South America
Accounting	Franca	South America
Direction	Franca	South America
Financial	São Paulo	South America
Sales	São Paulo	South America
Commercial	São Paulo	South America
Holder	São Paulo	South America

```
SELECT d.department_name, l.location_name, l.region_name
FROM department d, (
    SELECT c.country_name, r.region_name, l.location_city AS location_name,
    l.location_id
    FROM country c, region r, location l
    WHERE c.region_id = r.region_id
    AND l.country_id = c.country_id
    AND c.country_name = 'Brazil') AS l
WHERE d.location_id = l.location_id
ORDER BY l.location_name;
```

9 - Utilize uma subconsulta na cláusula HAVING a fim de mostrar apenas os funcionários que possuem salários superiores ao maior salário dos administradores. O relatório deverá exibir o nome completo dos funcionários em ordem alfabética, suas respectivas funções e salários.

Nome Completo	Função	Salário
Alcemar Illuminati	President	23300.00
Denis Snider	Counter	6584.66
Joana Souza	Analyst	5000.00
João Kipper	Counter	4444.05
Lucas Camargo	Engineer	8900.55
María Antonieta	Support	3000.00
Maurício Batista	Programmer	3500.00
Pedro Rocha	Director	16500.00
Valesca Mattos	Secretary	6000.00
Vanessa Hunemeier	Engineer	7500.00

```

SELECT e.employee_first_name || ' ' || e.employee_last_name AS "Nome Completo",
j.job_name AS "Função", MAX (e.employee_salary) AS "Salário"

FROM employee e INNER JOIN job j ON e.job_id = j.job_id

GROUP BY 1, 2

HAVING MAX (e.employee_salary) > (

    SELECT MAX(employee_salary)

    FROM employee ee, job jj

    WHERE ee.job_id = jj.job_id

    AND jj.job_name = 'Administrator')

ORDER BY 1;

```

10 - Utilize uma subconsulta na cláusula HAVING a fim de mostrar todos os empregados que possuem o cargo de administrador e salário superior à média salarial deste cargo.

Nome	Salário
Emilio Hennemann	2800.00

```

SELECT employee_first_name || ' ' || employee_last_name AS "Nome", employee_salary AS
"Salário"

FROM employee

WHERE job_id = 7

GROUP BY employee_first_name, employee_last_name, employee_salary

HAVING employee_salary > (

    SELECT ROUND (((job_max_salary - job_min_salary)/2),2)

    FROM job

    WHERE job_id = 7);

```

11 - Utilize uma subconsulta na cláusula HAVING a fim de mostrar o código dos departamentos que possuem uma média salarial superior ao teto salarial da TI.

Código do departamento	Média
10	15267
9	8200

```
SELECT department_id AS "Código do departamento", ROUND (AVG(employee_salary)) AS
"Média"
FROM employee
GROUP BY department_id
HAVING AVG(employee_salary) > (
    SELECT MAX(employee_salary)
    FROM employee
    WHERE department_id = 4);
```

12 - Utilize uma subconsulta na cláusula SELECT e subconsultas na cláusula HAVING a fim de mostrar o nome do empregado, quantas vezes o mesmo trocou de cargo e qual foi o seu primeiro cargo na empresa. Mostre somente os empregados que trocaram de cargo mais vezes que a média de troca de cargos dos empregados.

Nome do Empregado	Primeiro Cargo	Qtd de Cargos
Douglas Müller	Intern	2
Emilio Hennemann	Intern	3
Joana Souza	Secretary	2

```
SELECT e.employee_first_name || ' ' || e.employee_last_name AS "Nome do Empregado", (
    SELECT jb.job_name as "Primeiro Cargo"
    FROM job jb, job_history jh
    WHERE jb.job_id = jh.job_id
    AND MIN (j.job_history_start) = jh.job_history_start
    AND e.employee_id = jh.employee_id),
    COUNT (*) AS "Qtd de Cargos"
FROM job_history j, employee e, department d
WHERE d.department_id = e.department_id
AND j.employee_id = e.employee_id
GROUP BY e.employee_id, "Nome do Empregado", d.department_id
HAVING COUNT (*) > (
    SELECT AVG (c.qntd_cargos)
    FROM (
        SELECT COUNT (*) AS qntd_cargos
        FROM job_history j1, employee e1
```

```
WHERE j1.employee_id = e1.employee_id  
GROUP BY e1.employee_id) as c)  
ORDER BY "Nome do Empregado";
```