

Banco de dados II



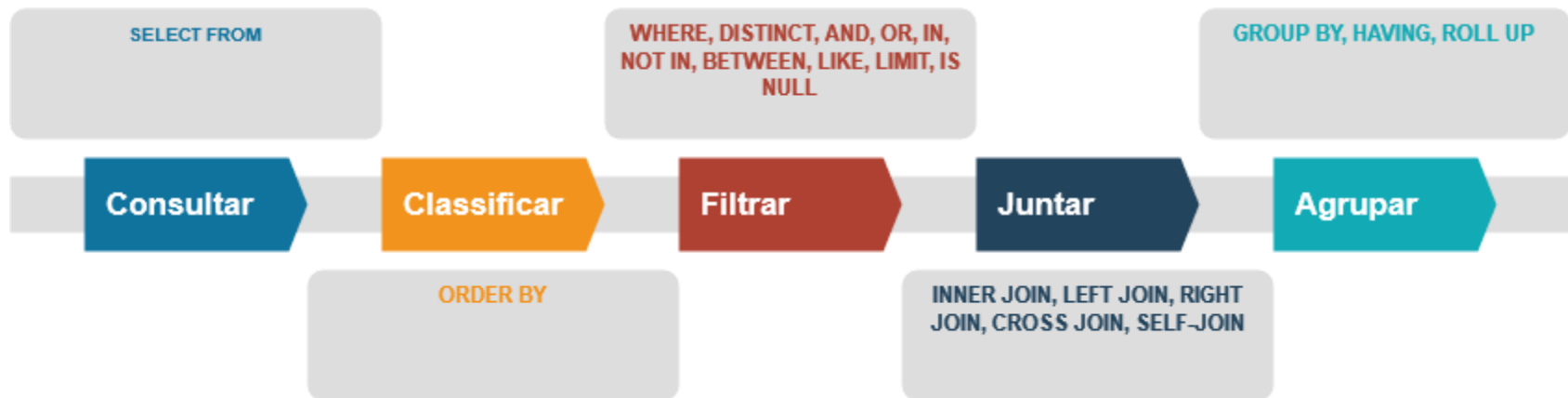
IFSP - Guarulhos

Objetivos

Estudar subqueries

Funções

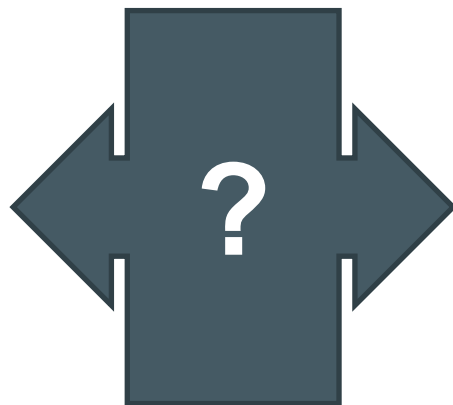
Revisando...



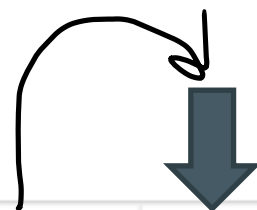
<https://www.mysqltutorial.org/mysql-basics/>

SQL CASE

employees
* employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
manager_id
department_id


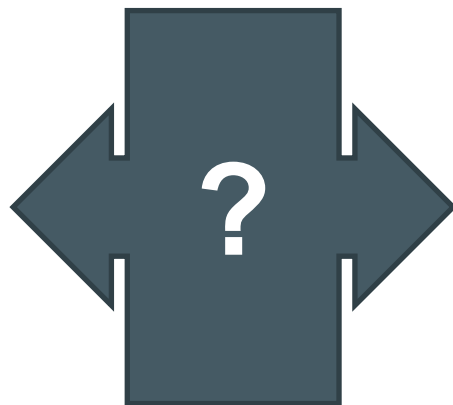


	first_name	last_name	hire_date	aniversary
▶	Adam	Fripp	1997-04-10	3 years
	Alexander	Hunold	1990-01-03	10 years
	Alexander	Khoo	1995-05-18	5 years
	Britney	Everett	1997-03-03	3 years
	Bruce	Ernst	1991-05-21	NULL
	Charles	Johnson	2000-01-04	NULL
	Daniel	Faviet	1994-08-16	NULL



SQL CASE

employees	
* employee_id	
first_name	
last_name	
email	
phone_number	
hire_date	
job_id	
salary	
manager_id	
department_id	



	first_name	last_name	salary	evaluation
►	Adam	Fripp	8200.00	High
	Alexander	Hunold	9000.00	High
	Alexander	Khoo	3100.00	Average
	Britney	Everett	3900.00	Average
	Bruce	Ernst	6000.00	High
	Charles	Johnson	6200.00	High
	Daniel	Faviet	9000.00	High
	David	Austin	4800.00	Average
	Den	Raphaely	11000.00	High
	Diana	Lorentz	4200.00	Average
	Guy	Himuro	2600.00	Low
	Hermann	Baer	10000.00	High
	Irene	Mikkilineni	2700.00	Low

Subqueries

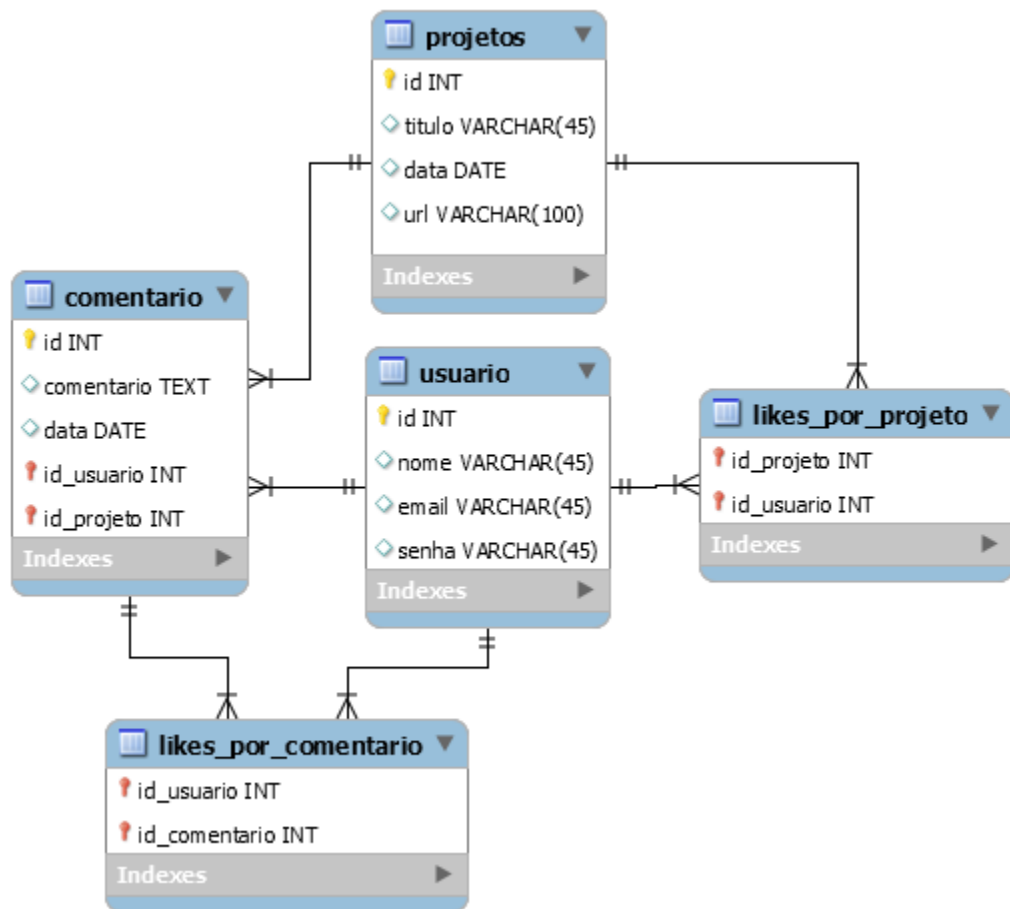
Subqueries

Uma Subquery (também conhecida como SUBCONSULTA ou SUBSELECT) é uma instrução do tipo SELECT dentro de outra instrução SQL, que efetua consultas que, de outra forma, seriam extremamente complicadas ou impossíveis de serem feitas.

<https://www.devmedia.com.br/trabalhando-com-subqueries/40134>

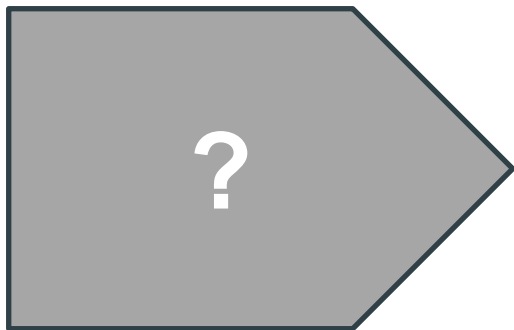
Como utilizar a subquery?

- **1) Subquery** como uma nova coluna da consulta (SELECT AS FIELD);
- **2) Subquery** como filtro de uma consulta (utilizando IN, EXISTS ou operadores de comparação);
- **3) Subquery** como fonte de dados de uma consulta principal (SELECT FROM SELECT).



**1 - Subquery como uma
nova coluna da consulta
(SELECT AS FIELD)**

1 - Subquery como uma nova coluna da consulta (**SELECT AS FIELD**);



titulo	Quantidade_Comentarios
Aplicação C#	2
Aplicação Ionic	3
Aplicação Python	2

1 - Subquery como uma nova coluna da consulta (**SELECT AS FIELD**);

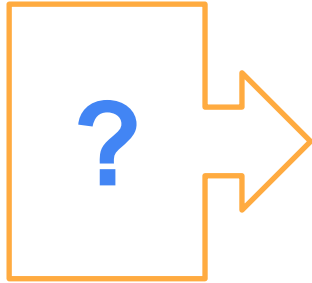


?

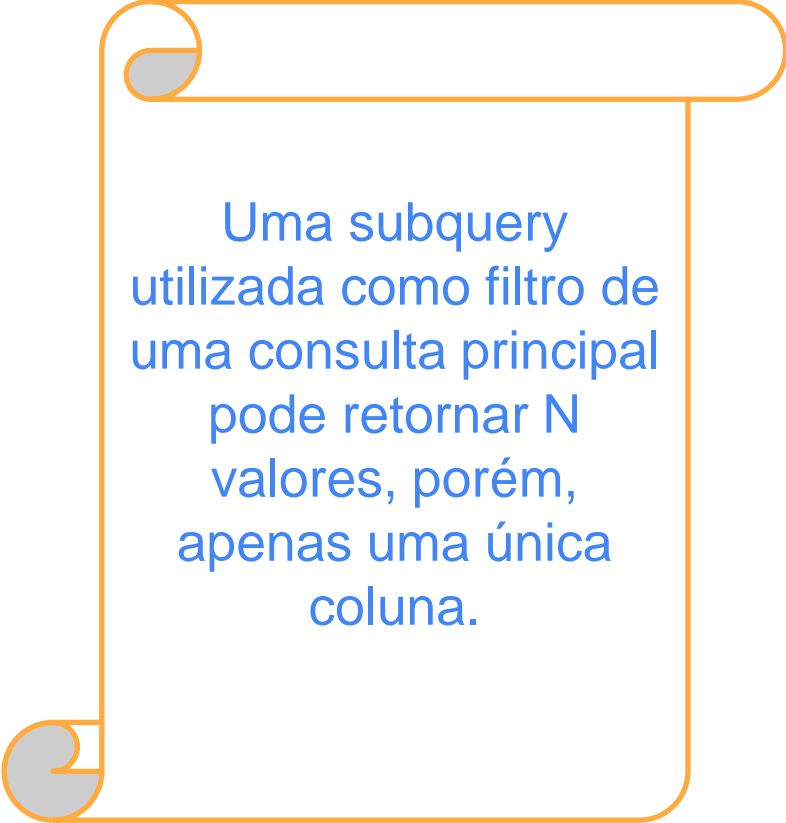
titulo	Quantidade_Comentarios	Quantidade_likes
Aplicação C#	2	2
Aplicação Ionic	3	4
Aplicação Python	2	1

2) Subquery como filtro de uma consulta (utilizando IN, EXISTS ou operadores de comparação)

2) Subquery como filtro de uma consulta (utilizando IN, EXISTS ou operadores de comparação);

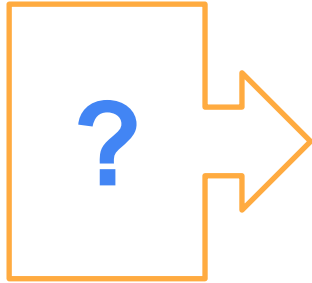


id	titulo	data
1	Aplicação C#	2018-04-01
2	Aplicação Ionic	2018-05-07
3	Aplicação Python	2018-08-05



Uma subquery
utilizada como filtro de
uma consulta principal
pode retornar N
valores, porém,
apenas uma única
coluna.

2) Subquery como filtro de uma consulta (utilizando IN, EXISTS ou operadores de comparação);

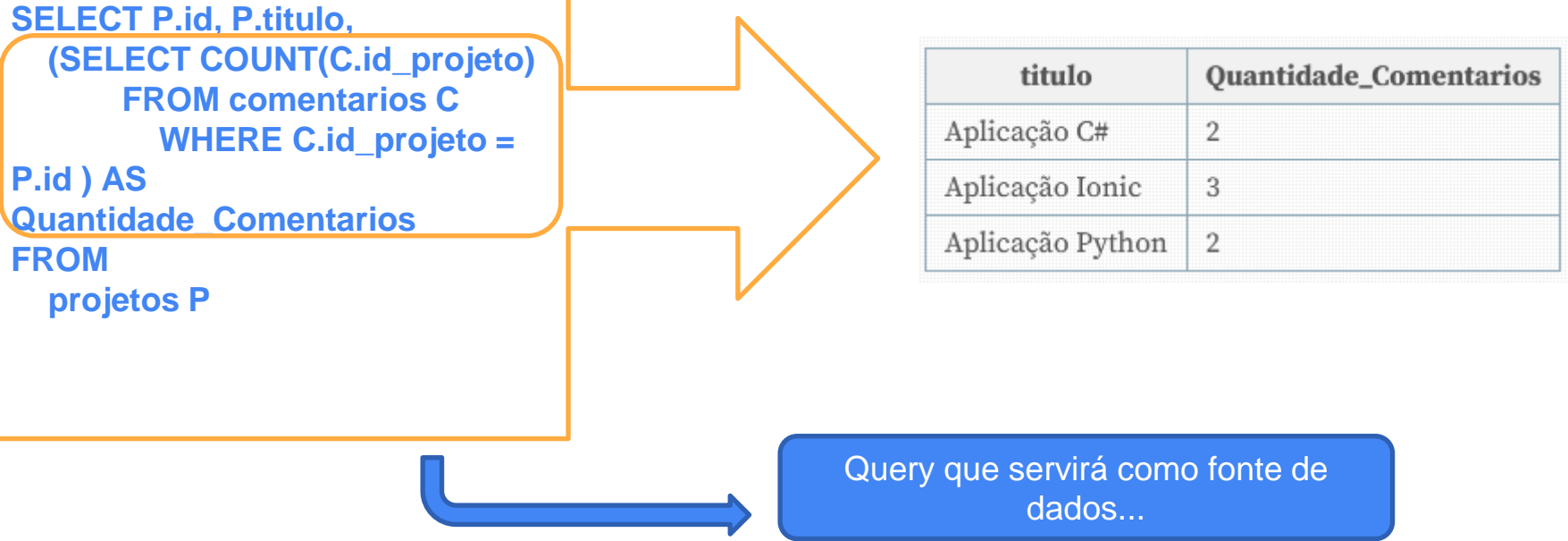


titulo	data
Aplicação Python	2018-08-05

3) Subquery como fonte de dados de uma consulta principal

3) Subquery como fonte de dados de uma consulta principal

```
SELECT P.id, P.titulo,  
       (SELECT COUNT(C.id_projeto)  
        FROM comentarios C  
        WHERE C.id_projeto =  
        P.id ) AS  
       Quantidade Comentarios  
FROM  
  projetos P
```



The diagram illustrates the execution of a SQL query. A large orange-bordered box on the left contains the SQL code. An orange arrow points from this box to a table on the right, representing the result of the query. A blue arrow points from the bottom of the orange box to a blue rounded rectangle at the bottom right, which contains the text 'Query que servirá como fonte de dados...'.

titulo	Quantidade_Comentarios
Aplicação C#	2
Aplicação Ionic	3
Aplicação Python	2

Query que servirá como fonte de dados...

3) Subquery como fonte de dados de uma consulta principal

```
SELECT P.id, P.titulo,  
       (SELECT  
        COUNT(C.id_projeto)  
        FROM comentarios C  
        WHERE C.id_projeto  
        = P.id ) AS  
       Quantidade_Comentarios  
FROM  
   projetos P
```

Query
que
servirá
como
fonte de
dados...

```
1  SELECT  
2      F.titulo,  
3      F.Quantidade_Comentarios  
4  FROM  
5      (  
6          SELECT  
7              P.id,  
8              P.titulo,  
9              (  
10                 SELECT  
11                     COUNT(C.id_projeto)  
12                 FROM  
13                     comentarios C  
14                 WHERE  
15                     C.id_projeto = P.id ) AS Quantidade_Comentarios  
16             ) as F  
17  FROM  
18      projetos P  
19  WHERE  
20      F.Quantidade_Comentarios > 2
```

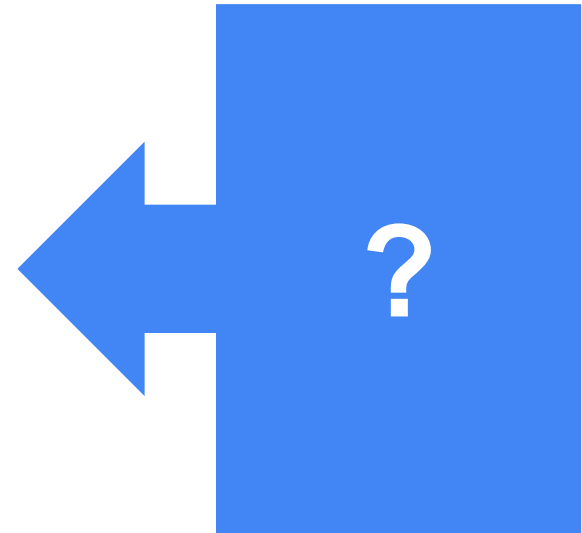
3) Subquery como fonte de dados de uma consulta principal

- Observe que na query do slide anterior a consulta principal solicita através do FROM as colunas **titulo**, **Quantidade_Comentarios** da fonte de dados baseada em uma outra consulta, e por fim, realiza um filtro no resultado através da cláusula WHERE para buscar somente aqueles projetos com a quantidade de comentários maior que 2.
- Sempre após a criação de uma subquery como fonte de dados de uma consulta principal será necessário definir um nome para esta fonte de dados, através da palavra reservada AS.

**Mais exemplos de
subqueries para
praticar...**

Subqueries

	department_id	department_name	location_id
▶	1	Administration	1700
	3	Purchasing	1700
	9	Executive	1700
	10	Finance	1700
	11	Accounting	1700



<https://www.sqltutorial.org/sql-sample-database/>

Ou para visualizar o modelo relacional:

<https://github.com/renatobdo/BD2/tree/main/semana2>

E se quisesse consultar...

- Todos os empregados em que o location_id seja igual a 1700? (Obs.: **SEM UTILIZAR INNER JOIN**)
- Poderíamos pegar o department_id que seja igual a 1, 3, 9, 10, 11 ?

	employee_id	first_name	last_name
►	115	Alexander	Khoo
	179	Charles	Johnson
	109	Daniel	Faviet
	114	Den	Raphaely
	118	Guy	Himuro
	111	Ismael	Sciarra
	177	Jack	Livingston
	200	Jennifer	Whalen
	110	John	Chen
	145	John	Russell

```
SELECT    employee_id, first_name, last_name
FROM      employees
WHERE     department_id IN
  (SELECT      department_id
   FROM        departments
   WHERE       location_id = 1700)
ORDER BY  first_name , last_name;
```



The diagram illustrates a subquery within an SQL statement. A blue rectangular box labeled "Subquery" has a large blue arrow pointing left towards a smaller orange rounded rectangle. This orange rectangle highlights the subquery portion of the SQL code: `(SELECT department_id FROM departments WHERE location_id = 1700)`. The main SQL statement is enclosed in a larger orange rounded rectangle.

Subquery

Qual empregado com maior salário?

```
SELECT employee_id, first_name, last_name, salary  
FROM employees  
WHERE salary =
```

**(SELECT MAX(salary)
FROM employees)**


```
ORDER BY first_name , last_name;
```



>
<
>=
<=
!= ou <>

EXISTE PELO MENOS UM... EXISTS


```
SELECT department_name  
FROM departments d  
WHERE EXISTS  
    ( SELECT 1  
      FROM employees e  
      WHERE salary > 10000 AND e.department_id = d.department_id)  
ORDER BY department_name;
```



Resultado
verdadeiro se a
subquery retorna
alguma linha

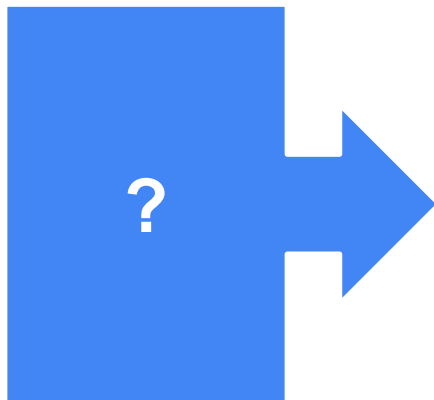
EXISTE PELO MENOS UM... EXISTS

```
SELECT department_name  
FROM departments d  
WHERE NOT EXISTS  
    ( SELECT 1  
      FROM employees e  
      WHERE salary > 10000 AND e.department_id = d.department_id)  
ORDER BY department_name;
```



Resultado verdadeiro
se a subquery retorna
alguma linha

Encontrando o menor salário por departamento



	MIN(salary)
▶	17000.00
	10000.00
	8300.00
	6900.00
	6500.00
	6200.00
	6000.00
	4400.00
	4200.00
	2700.00
	2500.00

Como encontrar todos os funcionários que possuem salário maior ou igual ao menor salário de cada departamento? ALL...



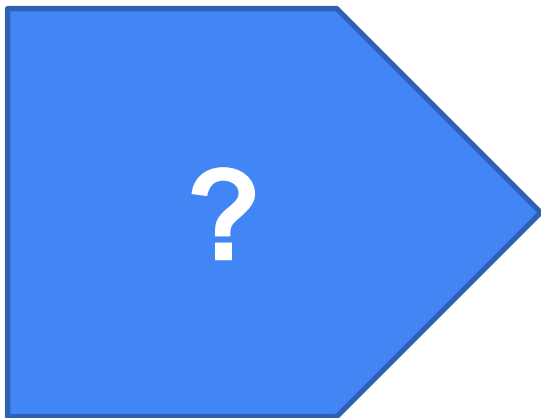
	employee_id	first_name	last_name	salary
▶	102	Lex	De Haan	17000.00
	101	Neena	Kochhar	17000.00
	100	Steven	King	24000.00

Como encontrar todos os funcionários que possuem salário maior ou igual ao maior salário de cada departamento? SOME...



	employee_id	first_name	last_name	salary
▶	121	Adam	Fripp	8200.00
	103	Alexander	Hunold	9000.00
	104	Bruce	Ernst	6000.00
	179	Charles	Johnson	6200.00
	109	Daniel	Faviet	9000.00
	105	David	Austin	4800.00
	114	Den	Raphaely	11000.00
	204	Hermann	Baer	10000.00
	111	Ismael	Sciarra	7700.00
	177	Jack	Livingston	8400.00
	200	Jennifer	Whalen	4400.00

Como obter a média salarial por departamento

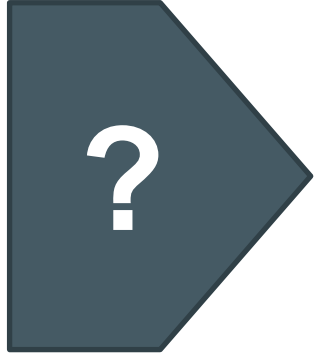


	average_salary
▶	4400.000000
	9500.000000
	4150.000000
	6500.000000
	5885.714286
	5760.000000
	10000.000000
	9616.666667
	19333.333333
	8600.000000
	10150.000000

Média das médias salariais

```
SELECT    ROUND(AVG(average_salary), 0)
FROM      (SELECT AVG(salary) average_salary
FROM employees
GROUP BY department_id) department_salary;
```


Subquery no SELECT



	employee_id	first_name	last_name	salary	average_salary	difference
▶	121	Adam	Fripp	8200.00	8060	140.00
	103	Alexander	Hunold	9000.00	8060	940.00
	115	Alexander	Khoo	3100.00	8060	-4960.00
	193	Britney	Everett	3900.00	8060	-4160.00
	104	Bruce	Ernst	6000.00	8060	-2060.00
	179	Charles	Johnson	6200.00	8060	-1860.00
	109	Daniel	Faviet	9000.00	8060	940.00
	105	David	Austin	4800.00	8060	-3260.00
	114	Den	Raphaely	11000.00	8060	2940.00
	107	Diana	Lorentz	4200.00	8060	-3860.00
	118	Guy	Himuro	2600.00	8060	-5460.00
	204	Hermann	Baer	10000.00	8060	1940.00

Introdução às funções

- <https://dev.mysql.com/doc/refman/8.0/en/create-procedure.html>
- <https://www.mysqltutorial.org/mysql-stored-function/>
- <https://www.techonthenet.com/mysql/functions.php>

Dúvidas projeto/atividade

Utilizem o final da aula para esclarecimento de dúvidas e desenvolvimento da atividade

Data de entrega dia 30/08/2022

Referências Bibliográficas

- <https://www.sqltutorial.org/sql-group-by/>
- <https://www.sqltutorial.org/sql-rollup/>
- <https://www.sqltutorial.org/sql-subquery/>
- <https://www.sqltutorial.org/sql-correlated-subquery/>