

## Operações Aritméticas e Operadores Lógicos e de Comparação no MySQL

Fonte: <https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html>  
<https://elias.praciano.com/2014/01/mysql-tipos-de-dados/>

O SQL permite executar **Operações Matemáticas** durante as consultas ao banco de dados; caso exista a necessidade de duas ou mais operações do mesmo procedimento (multiplicação e divisão ou adição e subtração), a operação **será realizada na ordem em que foi construída**; para alterar a prioridade, devemos utilizar parênteses.

+	Adição
-	Subtração
*	Multiplicação
/	Divisão

Para ampliarmos os filtros aplicados nas consultas temos os **Operadores Lógicos** que permitem refinar ainda mais os filtros aplicados:


AND	Retorna TRUE se ambas as condições forem verdadeiras
OR	Retorna TRUE se uma das condições for verdadeira
NOT	Retorna TRUE se a condição seguinte for falsa

Além disso podemos utilizar os **Operadores de Comparação** para estipular condições que comparam uma expressão a outro valor ou expressão; a tabela abaixo mostra os operadores:

=	Igual a
>	Maior que
>=	Maior ou igual a
<	Menor que
<=	Menor ou igual a
<>	Diferente de

## Preparando o ambiente

Vamos iniciar criando uma nova base de dados para que possamos exercitar as consultas; reforçando, segue algumas informações importantes que precisam ser observadas:

- **Todo comando executado deve ser, por boa prática, finalizado com “ponto e vírgula”;**
- **Atenção a sintaxe de cada procedimento; obedeça rigorosamente aos procedimentos abaixo, garantido assim que não ocorram distorções nas etapas do tutorial;**
- **Caso deseje aplicar apenas uma parte do script, selecione o mesmo com o mouse, destacando conforme imagens abaixo, e clique no botão Execute .**

No MySQL Workbench, abra uma nova Query TAB e execute a sequência abaixo:

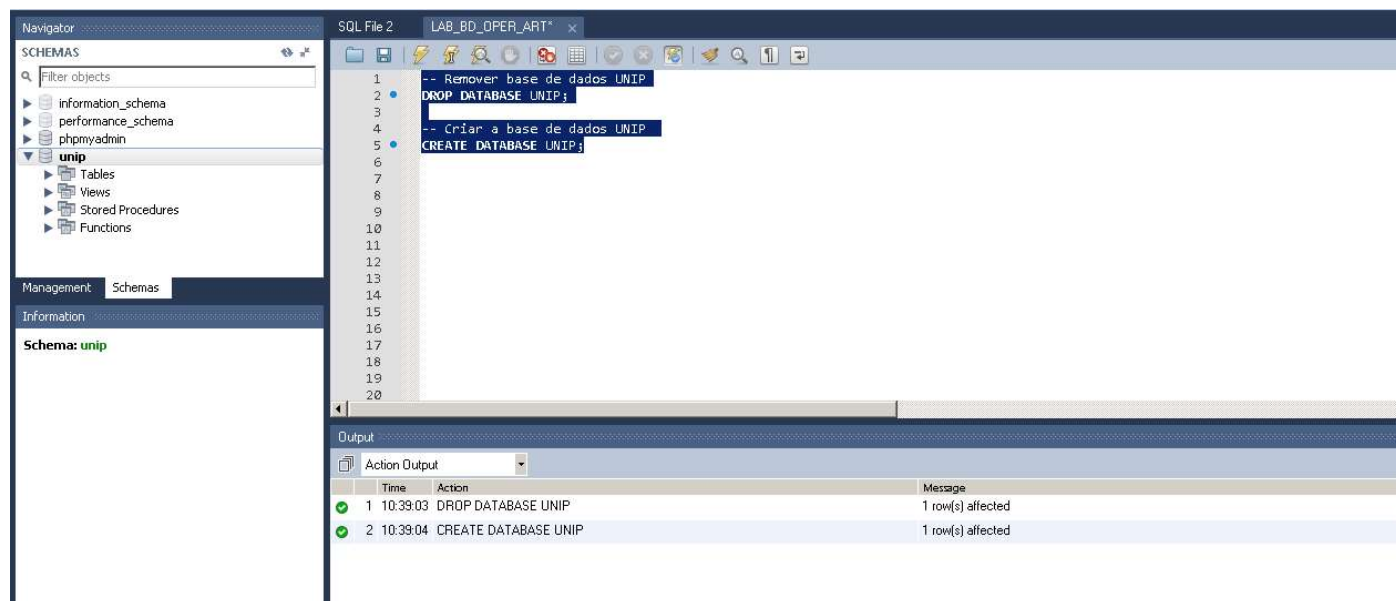
Para evitar problemas com a base utilizada na aula anterior, vamos iniciar recriando a base de dados UNIP; tenha cautela no uso do comando DROP pois este procedimento é irreversível:

```
-- Remover base de dados UNIP
```

```
DROP DATABASE UNIP;
```

```
-- Criar a base de dados UNIP
```

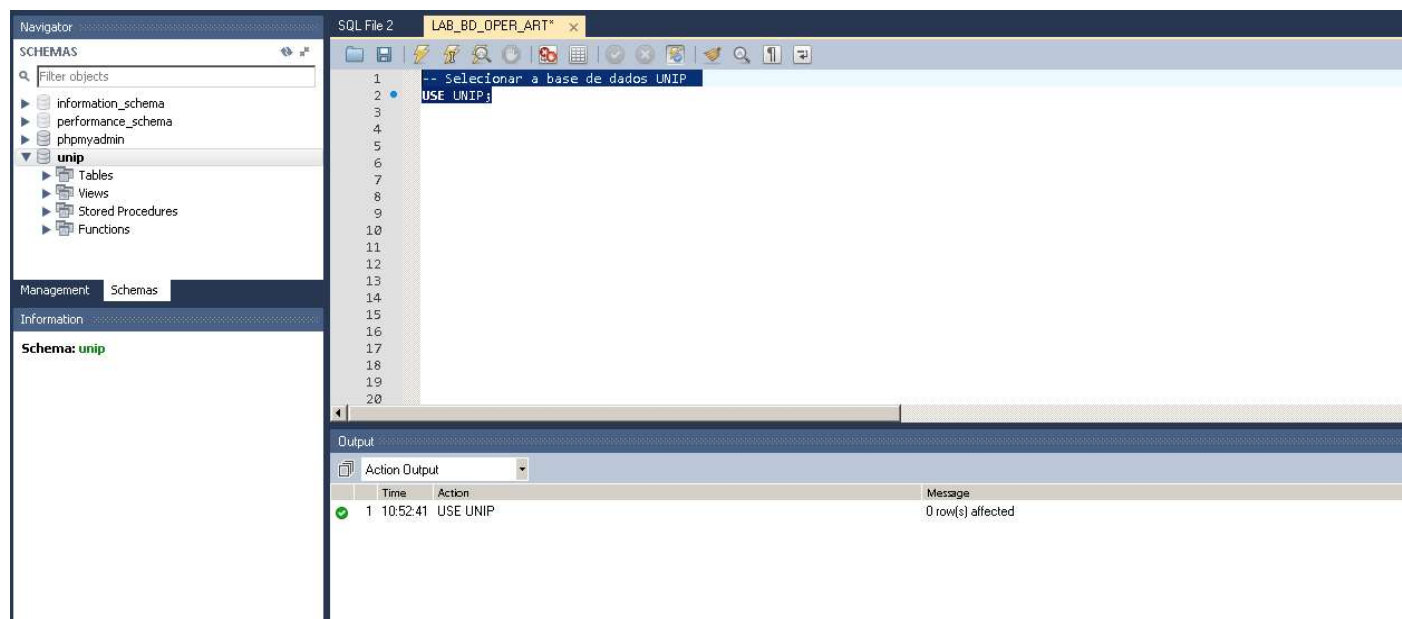
```
CREATE DATABASE UNIP;
```



Selecione a base de dados criada no passo anterior; veja que o BD em **NEGRITO** após ser selecionada:

```
-- Selecionar a base de dados UNIP
```

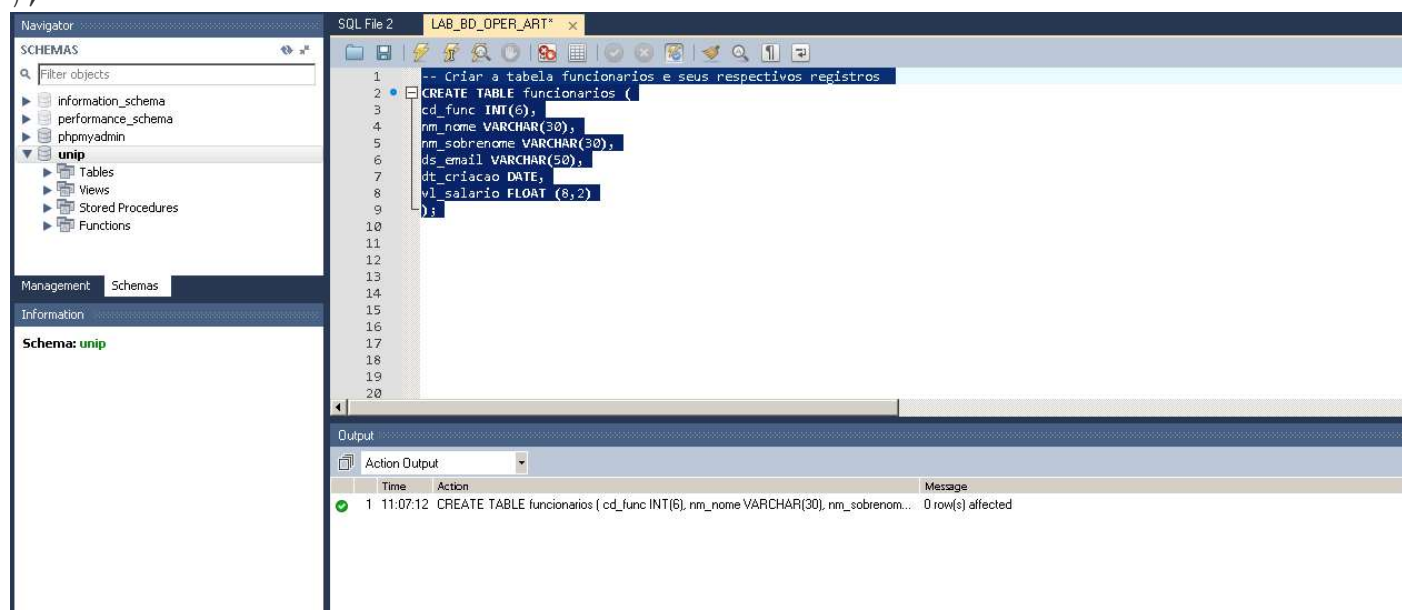
```
USE UNIP;
```



Crie a tabela funcionarios; **veja que neste primeiro momento não estamos tratando questões sobre chave primária e/ou estrangeira; estas características serão tratadas nas próximas aulas práticas:**

```
-- Criar a tabela funcionario e seus respectivos registros
```

```
CREATE TABLE funcionario (
    cd_func INT(6),
    nm_nome VARCHAR(30),
    nm_sobrenome VARCHAR(30),
    ds_email VARCHAR(50),
    dt_criacao DATE,
    vl_salario FLOAT (8,2)
);
```



Insira os dados abaixo na tabela criada; neste caso vamos adicionar um campo `vl_salario`, tipo `FLOAT`, que será usado nas consultas com os operadores aritméticos:

-- Inserindo registros na tabela funcionario

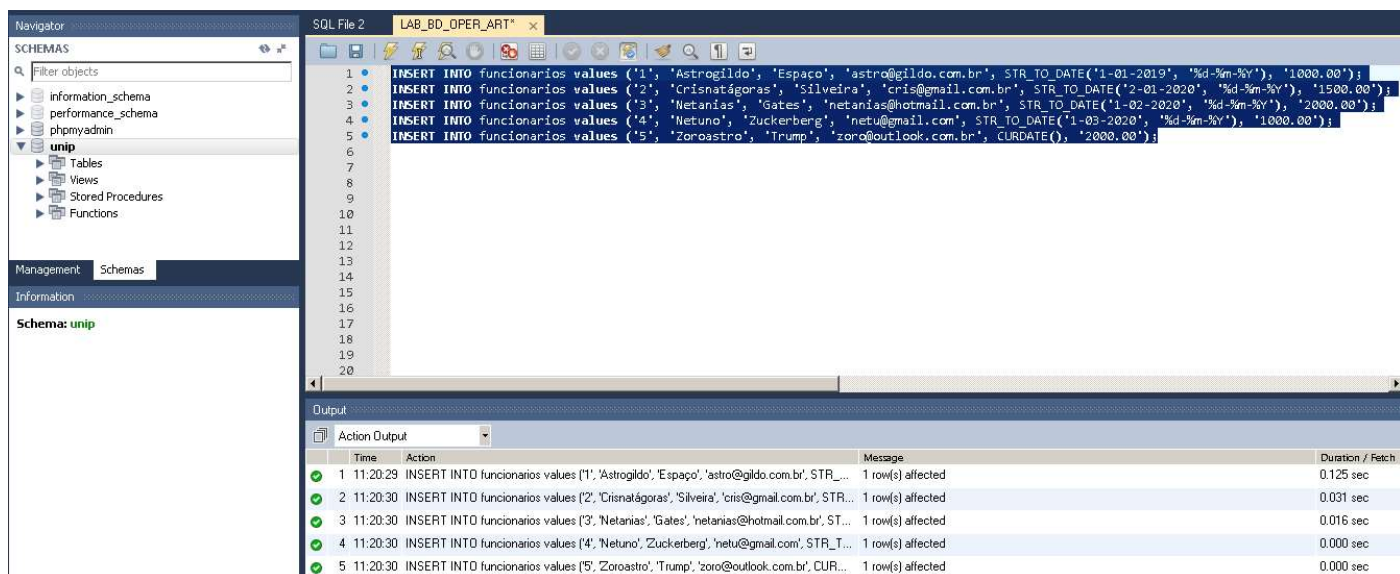
```
INSERT INTO funcionario values ('1', 'Astrogildo', 'Espaço', 'astro@gildo.com.br',
STR_TO_DATE('1-01-2019', '%d-%m-%Y'), '1000.00');

INSERT INTO funcionario values ('2', 'Crisnatágoras', 'Silveira', 'cris@gmail.com.br',
STR_TO_DATE('2-01-2020', '%d-%m-%Y'), '1500.00');

INSERT INTO funcionario values ('3', 'Netanias', 'Gates', 'netanias@hotmail.com.br',
STR_TO_DATE('1-02-2020', '%d-%m-%Y'), '2000.00');

INSERT INTO funcionario values ('4', 'Netuno', 'Zuckerberg', 'netu@gmail.com',
STR_TO_DATE('1-03-2020', '%d-%m-%Y'), '1000.00');

INSERT INTO funcionario values ('5', 'Zoroastro', 'Trump', 'zoro@outlook.com.br',
CURDATE(), '2000.00');
```

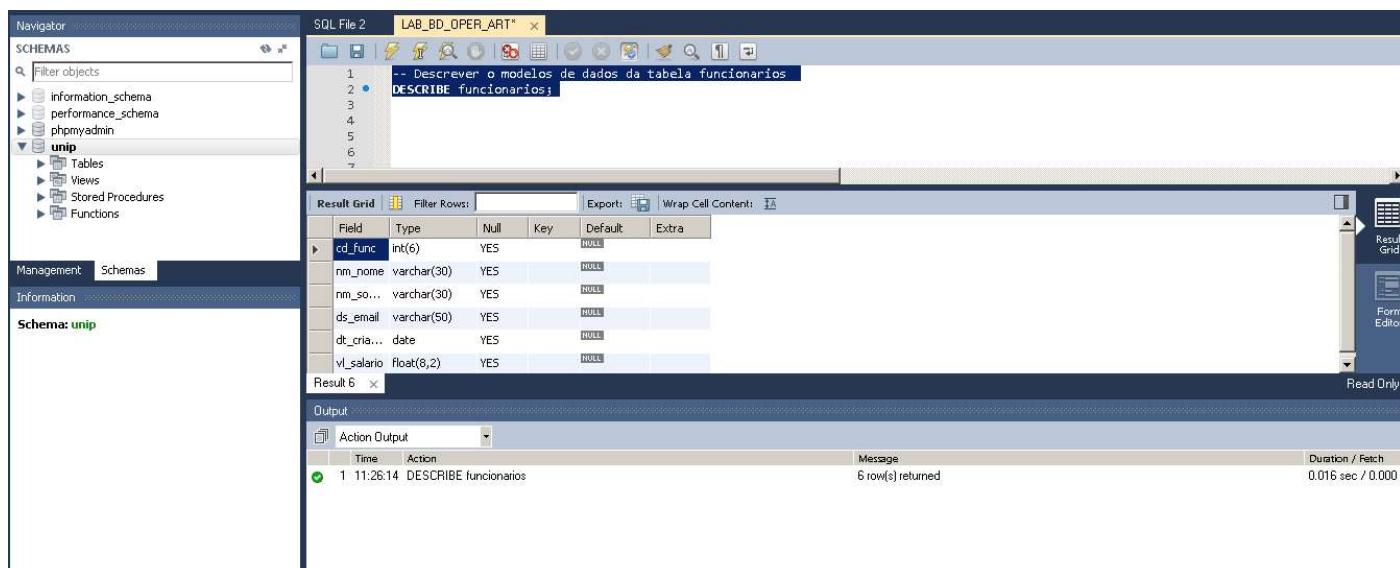


Time	Action	Message	Duration / Fetch
1 11:20:29	INSERT INTO funcionarios values ('1', 'Astrogildo', 'Espaço', 'astro@gildo.com.br', STR...	1 row(s) affected	0.125 sec
2 11:20:30	INSERT INTO funcionarios values ('2', 'Crisnatágoras', 'Silveira', 'cris@gmail.com.br', STR...	1 row(s) affected	0.031 sec
3 11:20:30	INSERT INTO funcionarios values ('3', 'Netanias', 'Gates', 'netanias@hotmail.com.br', ST...	1 row(s) affected	0.016 sec
4 11:20:30	INSERT INTO funcionarios values ('4', 'Netuno', 'Zuckerberg', 'netu@gmail.com', STR_T...	1 row(s) affected	0.000 sec
5 11:20:30	INSERT INTO funcionarios values ('5', 'Zoroastro', 'Trump', 'zoro@outlook.com.br', CUR...	1 row(s) affected	0.000 sec

Lembrando que é possível verificar a estrutura da sua tabela através do comando **DESCRIBE**:

-- Descrever o modelo de dados da tabela funcionario

DESCRIBE funcionarios;



Field	Type	Null	Key	Default	Extra
cd_func	int(6)	YES		NULL	
nm_nome	varchar(30)	YES		NULL	
nm_so...	varchar(30)	YES		NULL	
ds_email	varchar(50)	YES		NULL	
dt_cria...	date	YES		NULL	
vl_salario	float(8,2)	YES		NULL	

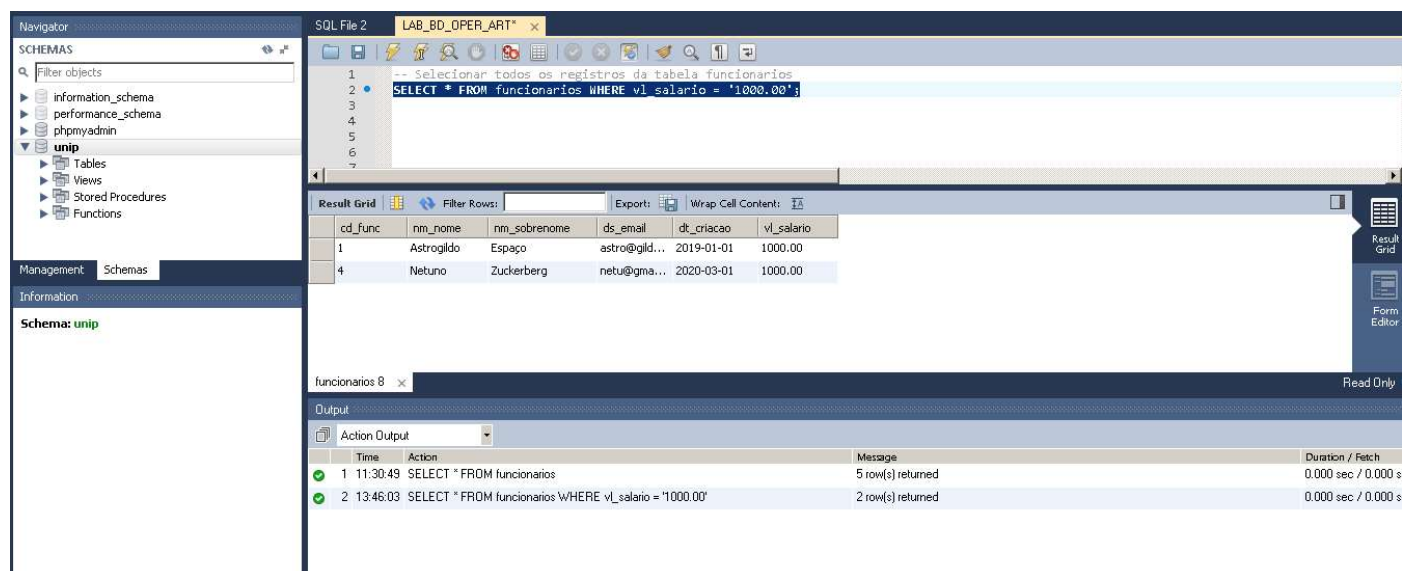
Time	Action	Message	Duration / Fetch
1 11:26:14	DESCRIBE funcionarios	6 row(s) returned	0.016 sec / 0.000 s

## Efetuating queries in MySQL using Arithmetic Operators

We will start by checking all the data in the table, independent of condition or filter; **perceba que neste SELECT utilizamos o Operador de Comparação “=” relacionado a nossa clausula WHERE:**

```
-- Selecionar todos os registros que possuem salario igual a R$ 1000,00
```

```
SELECT * FROM funcionario WHERE vl_salario = '1000.00';
```



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `-- Selecionar todos os registros da tabela funcionarios` and `SELECT * FROM funcionarios WHERE vl_salario = '1000.00';`. The Results window displays the following data:

cd_func	nm_nome	nm_sobrenome	ds_email	dt_criacao	vl_salario
1	Astrogildo	Espaco	astro@gild...	2019-01-01	1000.00
4	Netuno	Zuckerberg	netu@gma...	2020-03-01	1000.00

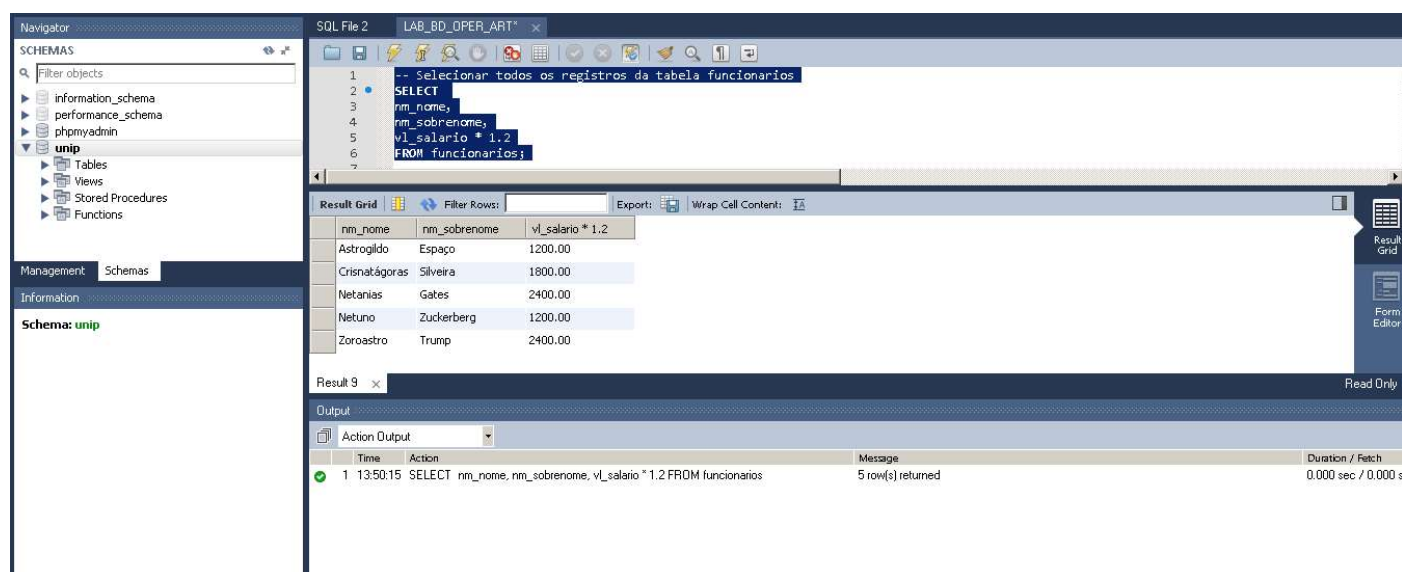
The Output window shows the execution log:

Time	Action	Message	Duration / Fetch
1 11:30:49	SELECT * FROM funcionarios	5 row(s) returned	0.000 sec / 0.000 s
2 13:46:03	SELECT * FROM funcionarios WHERE vl_salario = '1000.00'	2 row(s) returned	0.000 sec / 0.000 s

Now we will apply a percentage to all the salaries in the table:

```
-- Selecionar todos os registros da tabela funcionario, aplicando um aumento de 20%
```

```
SELECT
nm_nome,
nm_sobrenome,
vl_salario * 1.2
FROM funcionario;
```



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the query: `-- Selecionar todos os registros da tabela funcionarios` and `SELECT nm_nome, nm_sobrenome, vl_salario * 1.2 FROM funcionarios;`. The Results window displays the following data:

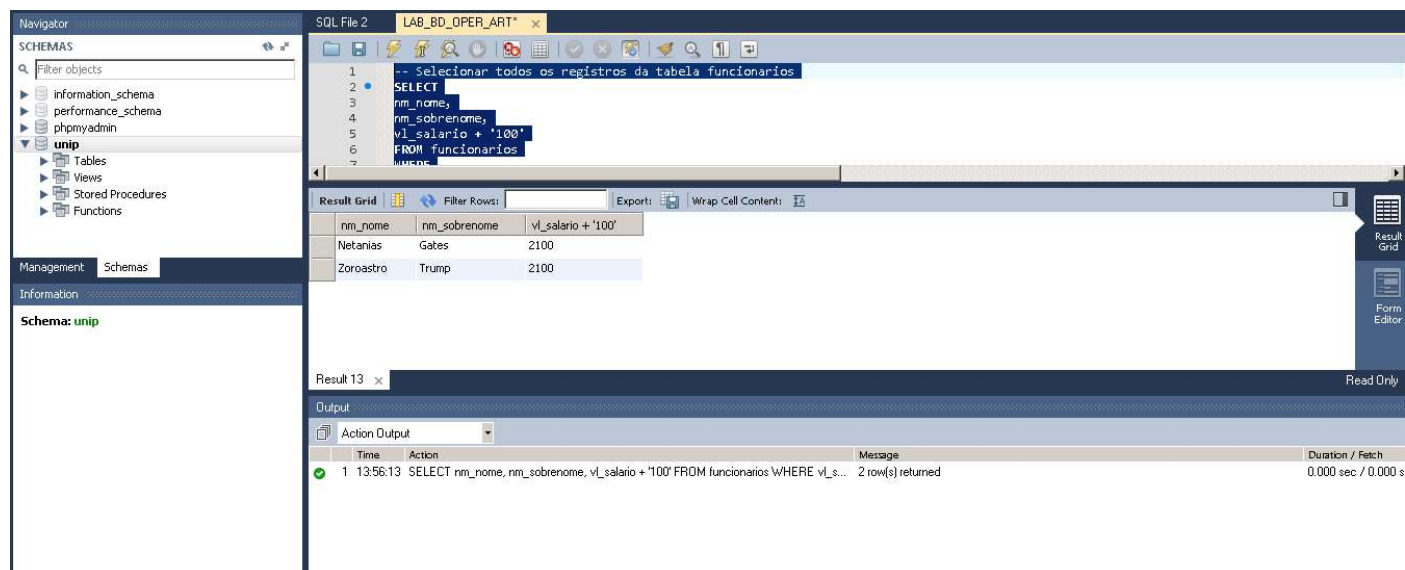
nm_nome	nm_sobrenome	vl_salario * 1.2
Astrogildo	Espaco	1200.00
Crisnatagoras	Silveira	1800.00
Netanias	Gates	2400.00
Netuno	Zuckerberg	1200.00
Zoroastro	Trump	2400.00

The Output window shows the execution log:

Time	Action	Message	Duration / Fetch
1 13:50:15	SELECT nm_nome, nm_sobrenome, vl_salario * 1.2 FROM funcionarios	5 row(s) returned	0.000 sec / 0.000 s

Neste comando vamos aplicar uma soma no valor de salário para o registro que atenda a condição WHERE;

```
-- Selecionar o registro conforme filtro e aplicar a soma de '100' ao valor de salario
SELECT
nm_nome,
nm_sobrenome,
vl_salario + '100'
FROM funcionario
WHERE
vl_salario = '2000.00';
```



The screenshot shows a MySQL IDE interface. On the left, the 'Navigator' pane displays the database schema 'unip'. The main editor shows a SQL query in a file named 'LAB\_BD\_OPER\_ART'. The query is as follows:

```
-- Selecionar todos os registros da tabela funcionarios
SELECT
nm_nome,
nm_sobrenome,
vl_salario + '100'
FROM funcionarios
WHERE
vl_salario = '2000.00';
```

Below the query editor, the 'Result Grid' displays the results of the query. The grid has three columns: 'nm\_nome', 'nm\_sobrenome', and 'vl\_salario + '100''. The results are as follows:

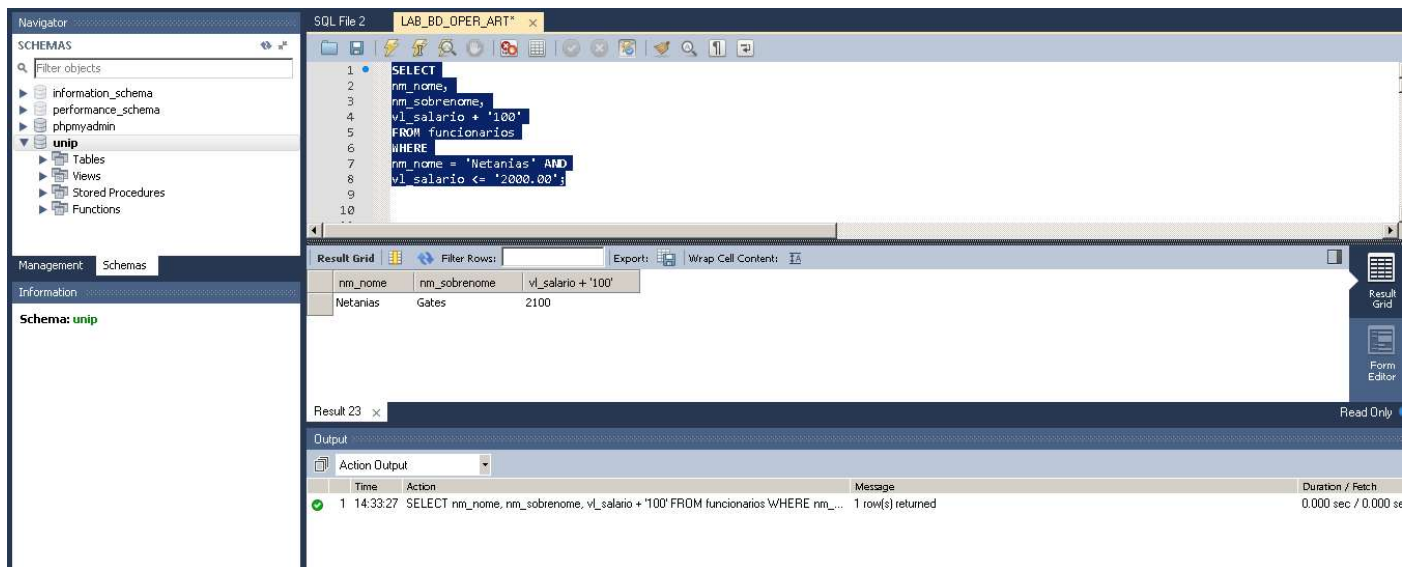
nm_nome	nm_sobrenome	vl_salario + '100'
Netanias	Gates	2100
Zoroastro	Trump	2100

At the bottom, the 'Output' pane shows the execution details for 'Result 13'. It indicates that the query was executed at 13:56:13 and returned 2 rows in 0.000 seconds.



Vamos repetir a consulta anterior, ampliando o filtro WHERE e alterando o Operador de Comparação:

```
-- Selecionar o registro conforme filtro e aplicar a soma de '100' ao valor de salario
SELECT
nm_nome,
nm_sobrenome,
vl_salario + '100'
FROM funcionario
WHERE
nm_nome = 'Netanias' AND
vl_salario <= '2000.00';
```



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
SELECT
nm_nome,
nm_sobrenome,
vl_salario + '100'
FROM funcionario
WHERE
nm_nome = 'Netanias' AND
vl_salario <= '2000.00';
```

The Results window displays the following data:

nm_nome	nm_sobrenome	vl_salario + '100'
Netanias	Gates	2100

The Output window shows the following message:

```
1 14:33:27 SELECT nm_nome, nm_sobrenome, vl_salario + '100' FROM funcionario WHERE nm_nome = 'Netanias' AND vl_salario <= '2000.00'; 1 row(s) returned 0.000 sec / 0.000 sec
```

Utilizando o ALIAS, vamos aplicar outro Operador Lógico no filtro:

-- Selecionar o registro conforme filtro e aplicar o aumento de 15% no valor de salario

SELECT

f.cd\_func,

f.nm\_nome,

f.nm\_sobrenome,

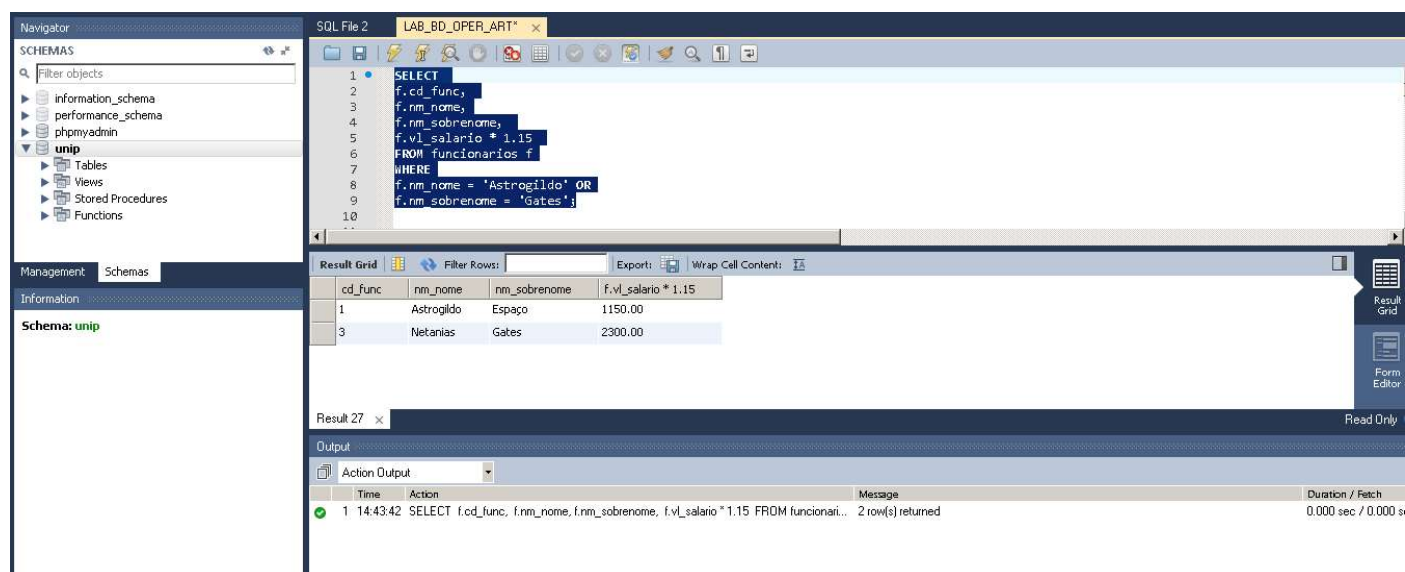
f.vl\_salario \* 1.15

FROM funcionario f

WHERE

f.nm\_nome = 'Astrogildo' OR

f.nm\_sobrenome = 'Gates';



SQL File 2 LAB\_BD\_OPER\_ART

```

1 SELECT
2 f.cd_func,
3 f.nm_nome,
4 f.nm_sobrenome,
5 f.vl_salario * 1.15
6 FROM funcionarios f
7 WHERE
8 f.nm_nome = 'Astrogildo' OR
9 f.nm_sobrenome = 'Gates';
10

```

cd_func	nm_nome	nm_sobrenome	f.vl_salario * 1.15
1	Astrogildo	Espago	1150.00
3	Netanias	Gates	2300.00

Result 27 x

Output

Action Output

Time	Action	Message	Duration / Fetch
1 14:43:42	SELECT f.cd_func, f.nm_nome, f.nm_sobrenome, f.vl_salario * 1.15 FROM funcionarios...	2 row(s) returned	0.000 sec / 0.000 s



Repita o comando anterior, alterando o Operador Lógico e adicionado o Operador de Comparação:

-- Selecionar o registro conforme filtro e aplicar o aumento de 5% no valor de salario

SELECT

f.cd\_func,

f.nm\_nome,

f.nm\_sobrenome,

f.vl\_salario \* 1.05

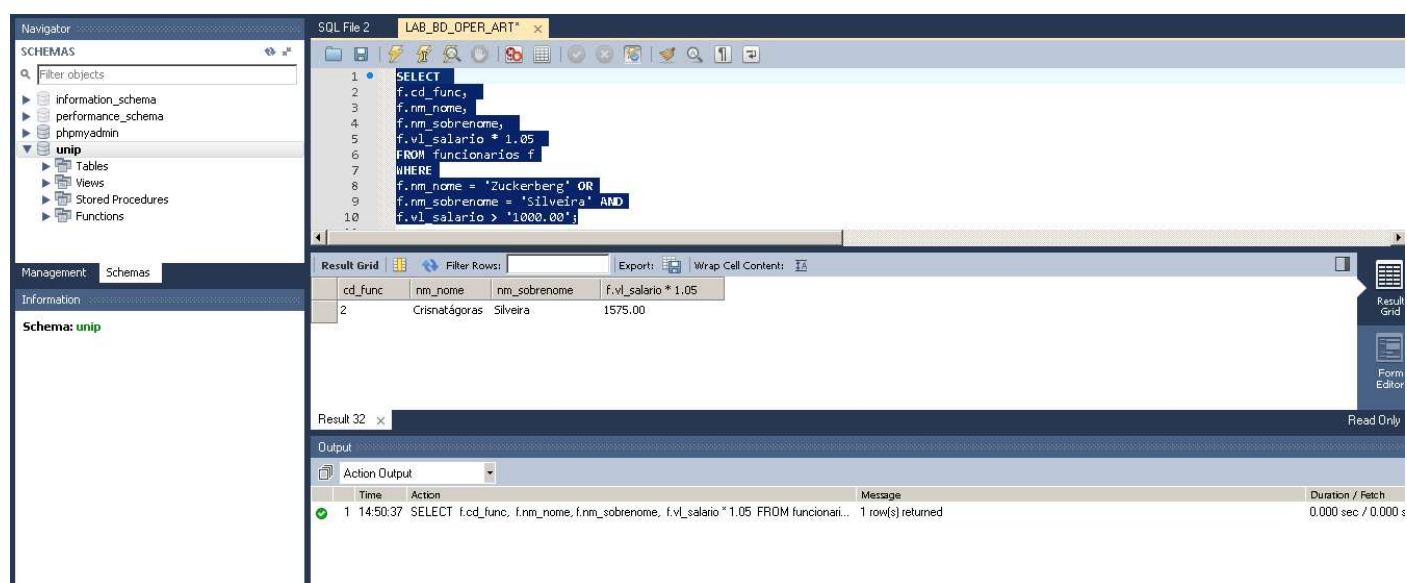
FROM funcionario f

WHERE

f.nm\_nome = 'Zuckerberg' OR

f.nm\_sobrenome = 'Silveira' AND

f.vl\_salario > '1000.00';



The screenshot shows the MySQL Workbench interface. The SQL Editor window displays the following query:

```

1 SELECT
2   f.cd_func,
3   f.nm_nome,
4   f.nm_sobrenome,
5   f.vl_salario * 1.05
6 FROM funcionarios f
7 WHERE
8   f.nm_nome = 'Zuckerberg' OR
9   f.nm_sobrenome = 'Silveira' AND
10  f.vl_salario > '1000.00';

```

The Results window shows the following data:

cd_func	nm_nome	nm_sobrenome	f.vl_salario * 1.05
2	Crisnatógoras	Silveira	1575.00

The Action Output window shows the following message:

```

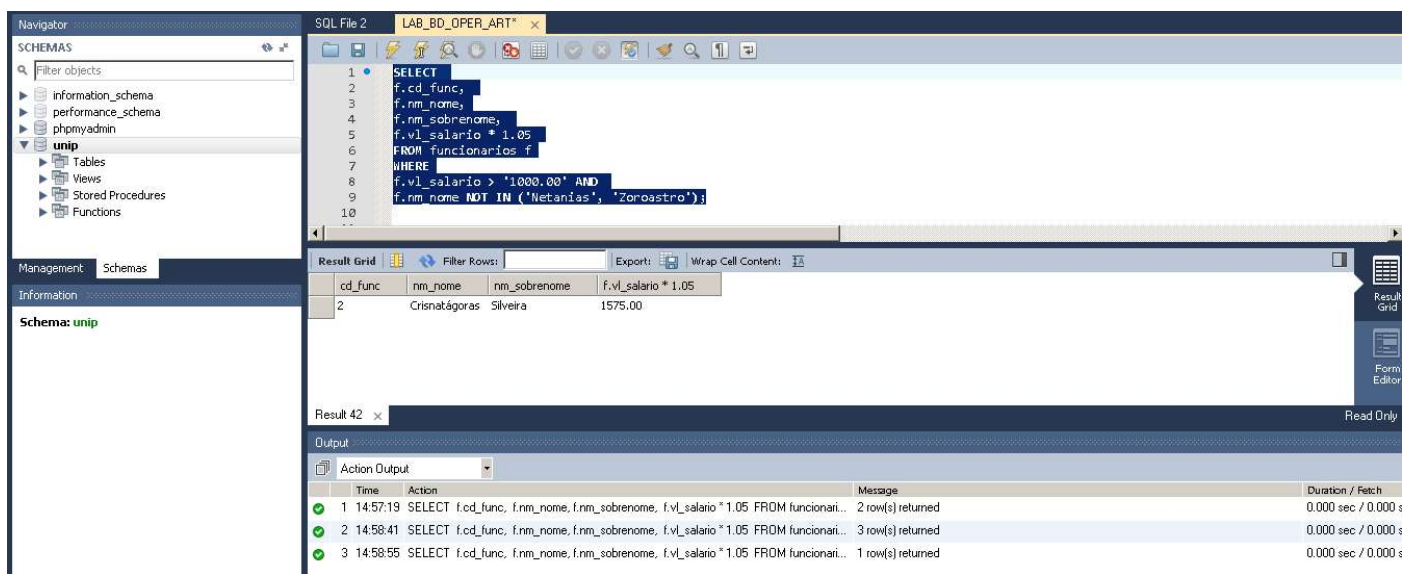
1 14:50:37 SELECT f.cd_func, f.nm_nome, f.nm_sobrenome, f.vl_salario * 1.05 FROM funcionarios... 1 row(s) returned
0.000 sec / 0.000 s

```

Por fim, vamos utilizar o Operador Lógico NOT concatenado com a clausula IN; nesta consulta você pode especificar múltiplas condições para um mesmo campo, neste caso, valores que o SELECT não deve considerar no resultado da query:

```
-- Selecionar o registro conforme filtro, aplicando o operador NOT IN
```

```
SELECT
f.cd_func,
f.nm_nome,
f.nm_sobrenome,
f.vl_salario
FROM funcionario f
WHERE
f.vl_salario > '1000.00' AND
f.nm_nome NOT IN ('Netanias', 'Zoroastro');
```



The screenshot shows a MySQL IDE interface. The SQL Editor displays the query: `SELECT f.cd_func, f.nm_nome, f.nm_sobrenome, f.vl_salario * 1.05 FROM funcionarios f WHERE f.vl_salario > '1000.00' AND f.nm_nome NOT IN ('Netanias', 'Zoroastro');`. The Results Grid shows one row with the following data: 

cd_func	nm_nome	nm_sobrenome	f.vl_salario * 1.05
2	Crisnatógoras	Silveira	1575.00

. The Action Output pane shows the execution log with three entries, all successful.

## Exercícios

- 1- Utilizando um apelido para a tabela funcionarios, efetue uma consulta que retorne todos os registros que possuem código maior que 2 e salário menor que R\$ 1500,00.
- 2- Utilizando um apelido para a tabela funcionarios, efetue uma consulta que retorne todos os registros que possuem nome igual a 'Netuno' ou sobrenome igual a 'Trump'.
- 3- Utilizando Operadores de Comparação, efetue uma consulta que retorne todos os registros com salário diferente de R\$ 1000,00.
- 4- Utilizando Operadores de Comparação, efetue uma consulta que retorne todos os registros com salário igual ou maior que R\$ 1500,00.
- 5- Efetue uma consulta que retorne o salário do funcionário atual do funcionário, o salário com aumento de 20% e o salário, já com o aumento de 20%, porém, descontado do imposto de 10%.