

Funções de agregação do SQL

Fonte: <http://www.bosontreinamentos.com.br/mysql/mysql-funcoes-de-agregacao-max-min-avg-count-sum-18/>

Como sabemos, o SQL – *Structured Query Language* (Linguagem de Consulta Estruturada) é uma linguagem usada para dar manutenção nos dados em um bancos de dados; os servidores de bancos de dados se valem dessa linguagem tanto para permitir a extração dos dados armazenados como para inserir novos dados ou atualizar aqueles já existentes.

Desde a primeira versão, o padrão incorpora funções classificadas como “agregação”; são funções nativas de consulta que servem para agrupar informações, agregando numa única linha o conteúdo de vários registros. Dessa forma, os detalhes da informação original são postos de lado e as informações são tratadas em conjunto.


Segue algumas das funções de agregação disponíveis no SQL:

COUNT	Retornar a quantidade de registros existentes
AVG	Calcular a média dos valores de um campo determinado
SUM	Calcular a soma dos valores de um determinado campo
MAX	Retornar o maior valor existente para um campo especificado
MIN	Retornar o menor valor existente para um campo especificado

Todas essas funções são aplicadas apenas aos registros que atenderem os critérios estabelecidos na cláusula WHERE da query onde estiverem sendo usadas.

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 os passos 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.

Caso esteja utilizado o PHPMyAdmin no servidor Azure desconsidere as etapas que envolvem remoção/criação da base de dados.

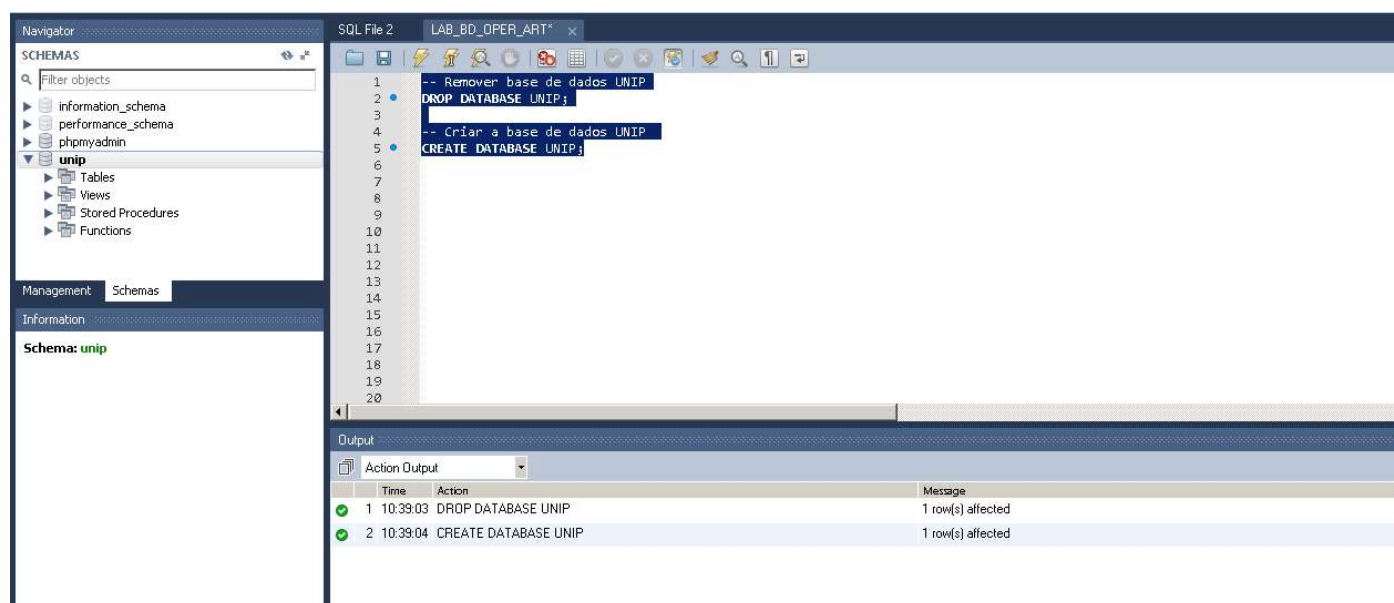
Para evitar problemas com a base utilizada na aula anterior, vamos iniciar recriando a base de dados UNIP; tenha cautela no uso do comando DROP pois este procedimento é irreversível:

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

```
DROP DATABASE UNIP;
```

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

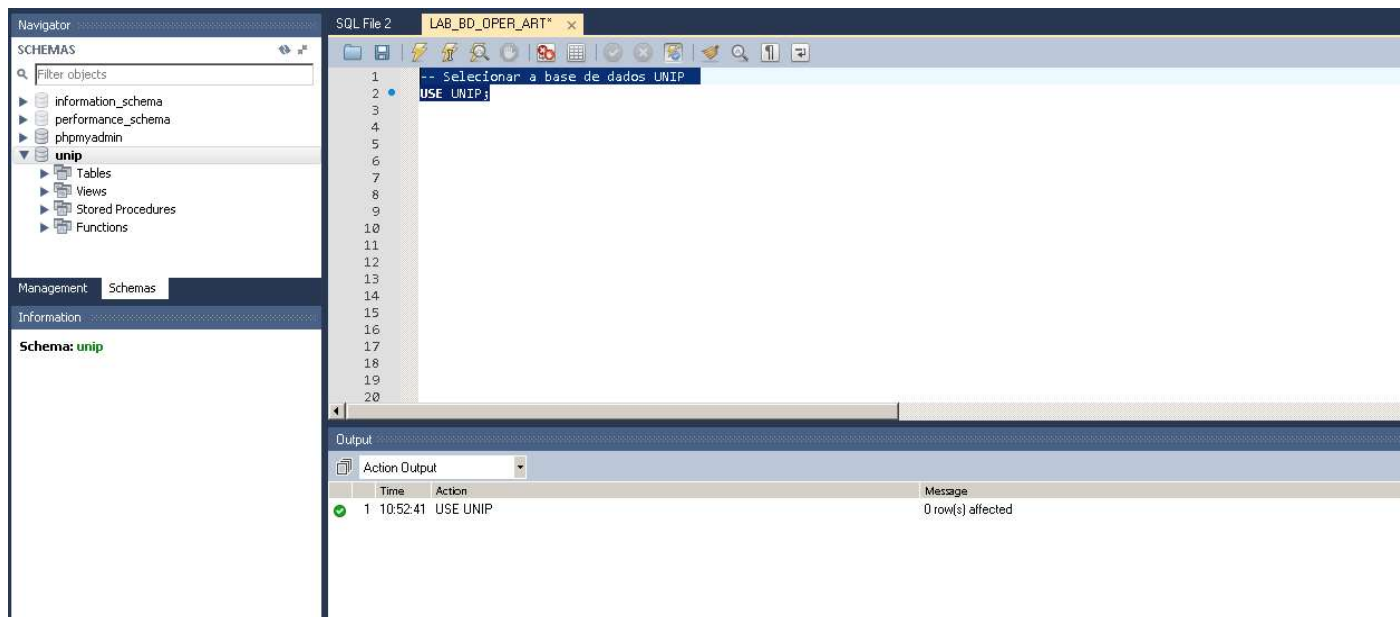
```
CREATE DATABASE UNIP;
```



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;
```



The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'unip' database selected under 'Schemas'. The main editor, 'SQL File 2', contains the command:


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

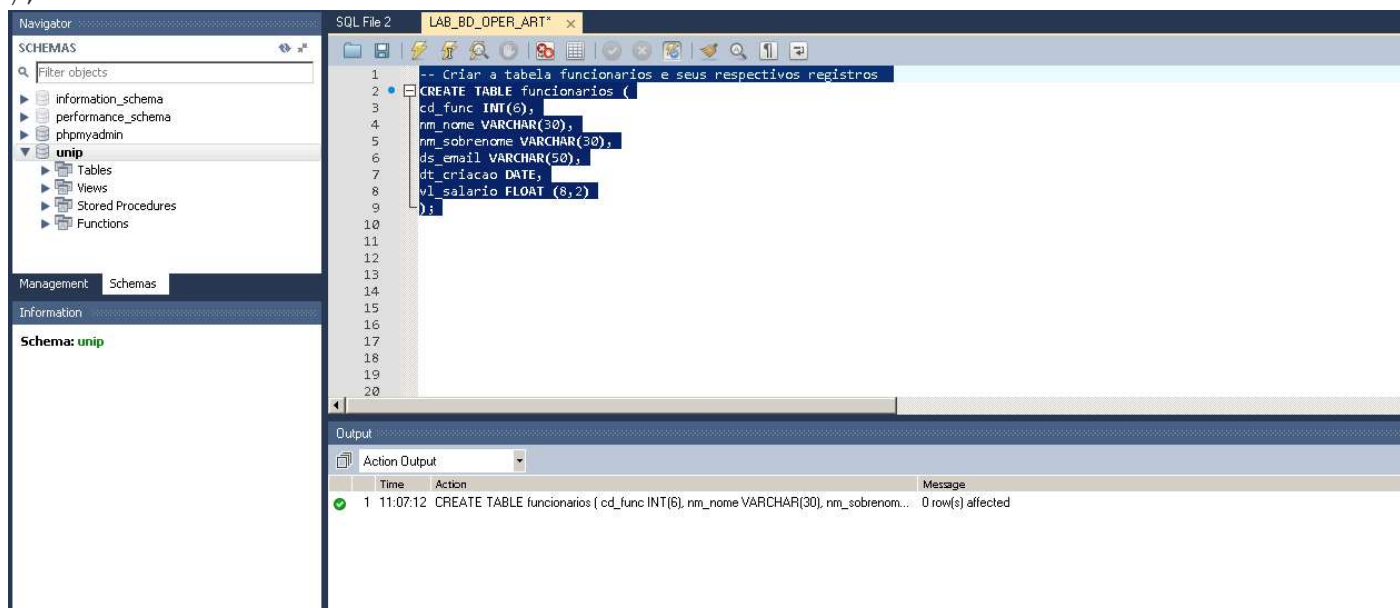
 The 'Output' pane at the bottom shows the execution result:

Time	Action	Message
1 10:52:41	USE UNIP	0 row(s) affected

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 funcionarios e seus respectivos registros
```

```
CREATE TABLE funcionarios (
  cd_func INT,
  nm_nome VARCHAR(30),
  nm_sobrenome VARCHAR(30),
  ds_email VARCHAR(50),
  dt_criacao DATE,
  vl_salario FLOAT
);
```



The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the 'unip' database selected under 'Schemas'. The main editor, 'SQL File 2', contains the command:


```
-- Criar a tabela funcionarios e seus respectivos registros
CREATE TABLE funcionarios (
  cd_func INT(6),
  nm_nome VARCHAR(30),
  nm_sobrenome VARCHAR(30),
  ds_email VARCHAR(50),
  dt_criacao DATE,
  vl_salario FLOAT (8,2)
);
```

 The 'Output' pane at the bottom shows the execution result:

Time	Action	Message
1 11:07:12	CREATE TABLE funcionarios (cd_func INT(6), nm_nome VARCHAR(30), nm_sobrenom...	0 row(s) affected

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 funcionarios

```
INSERT INTO funcionarios values ('1', 'Astrogildo', 'Espaço', 'astro@gildo.com.br',
'2019-01-01', '1000.00');
```

```
INSERT INTO funcionarios values ('2', 'Crisnatágoras', 'Silveira', 'cris@gmail.com.br',
'2020-01-02', '1500.00');
```

```
INSERT INTO funcionarios values ('3', 'Netanias', 'Gates', 'netanias@hotmail.com.br',
'2020-02-01', '2000.00');
```

```
INSERT INTO funcionarios values ('4', 'Netuno', 'Zuckerberg', 'netu@gmail.com', '2020-
03-01', '1000.00');
```

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

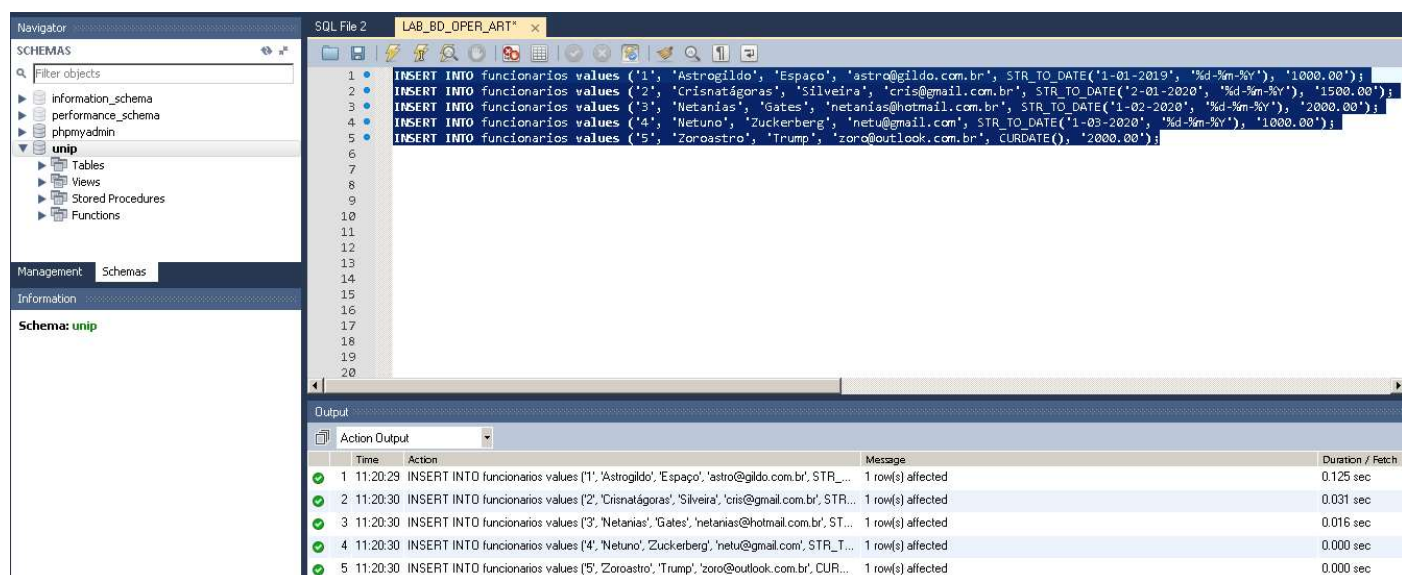
```
INSERT INTO funcionarios values ('6', 'Asdrubal', 'Maia', 'asdru@outlook.com.br',
CURDATE(), '10000.00');
```

```
INSERT INTO funcionarios values ('7', 'Astrogildo', 'Silva', 'a.silva@email.com',
CURDATE(), '10500.00');
```

```
INSERT INTO funcionarios values ('8', 'Bernardo', 'Santos', 'berna_123@hotmail.com',
CURDATE(), '12500.00');
```

```
INSERT INTO funcionarios (cd_func, nm_nome, nm_sobrenome, ds_email, dt_criacao) values
('9', 'Carlota', 'Joaquina', 'car_joa@gmail.com.br', CURDATE());
```

```
INSERT INTO funcionarios (cd_func, nm_nome, ds_email, dt_criacao, vl_salario) values
('10', 'Denilson', 'denao_ota@gmail.com.br', CURDATE(), '20500.00');
```



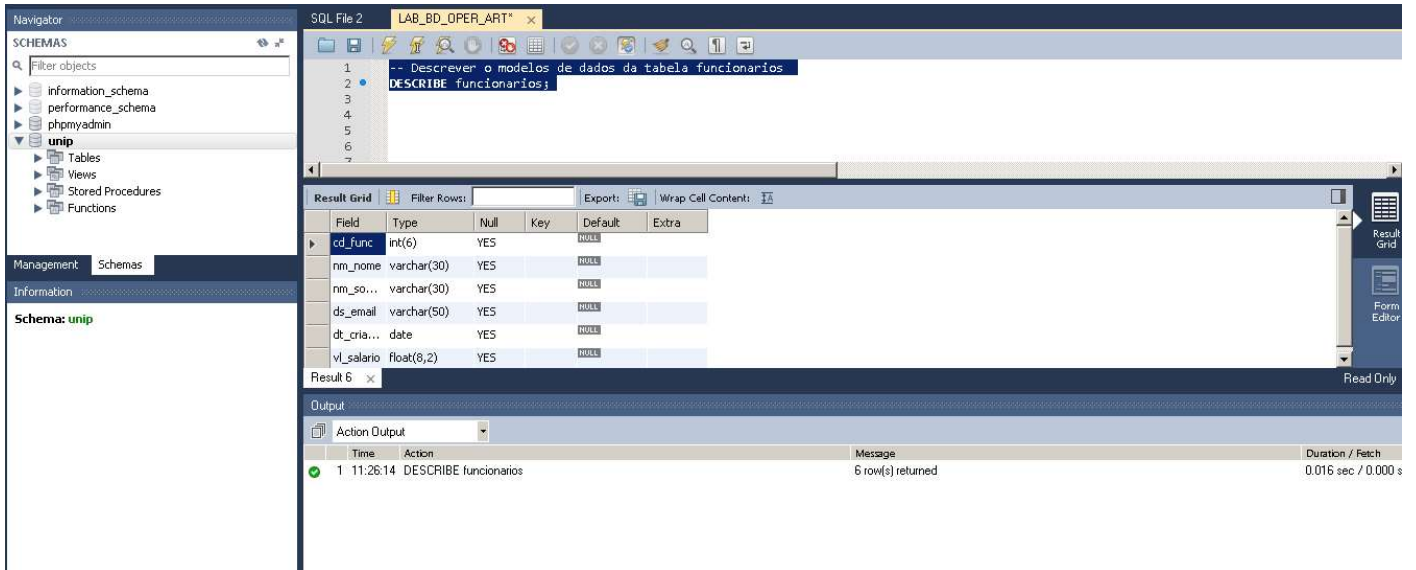
The screenshot shows a SQL IDE interface with a schema browser on the left, a SQL editor in the center, and an output window at the bottom. The schema browser shows a database named 'unip' with various tables and views. The SQL editor contains five INSERT statements for the 'funcionarios' table. The output window shows the execution results of these statements, indicating that each statement successfully affected one row.

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 funcionarios
```

```
DESCRIBE funcionarios;
```



The screenshot shows a SQL IDE interface with the following components:

- Navigator:** Displays a tree view of schemas including 'information_schema', 'performance_schema', 'phpmyadmin', and 'unip'. The 'unip' schema is selected, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'.
- SQL File 2:** Contains the SQL command:

```
-- Descrever o modelo de dados da tabela funcionarios
DESCRIBE funcionarios;
```
- Result Grid:** Displays the structure of the 'funcionarios' table:

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	
- Output:** Shows the execution results:

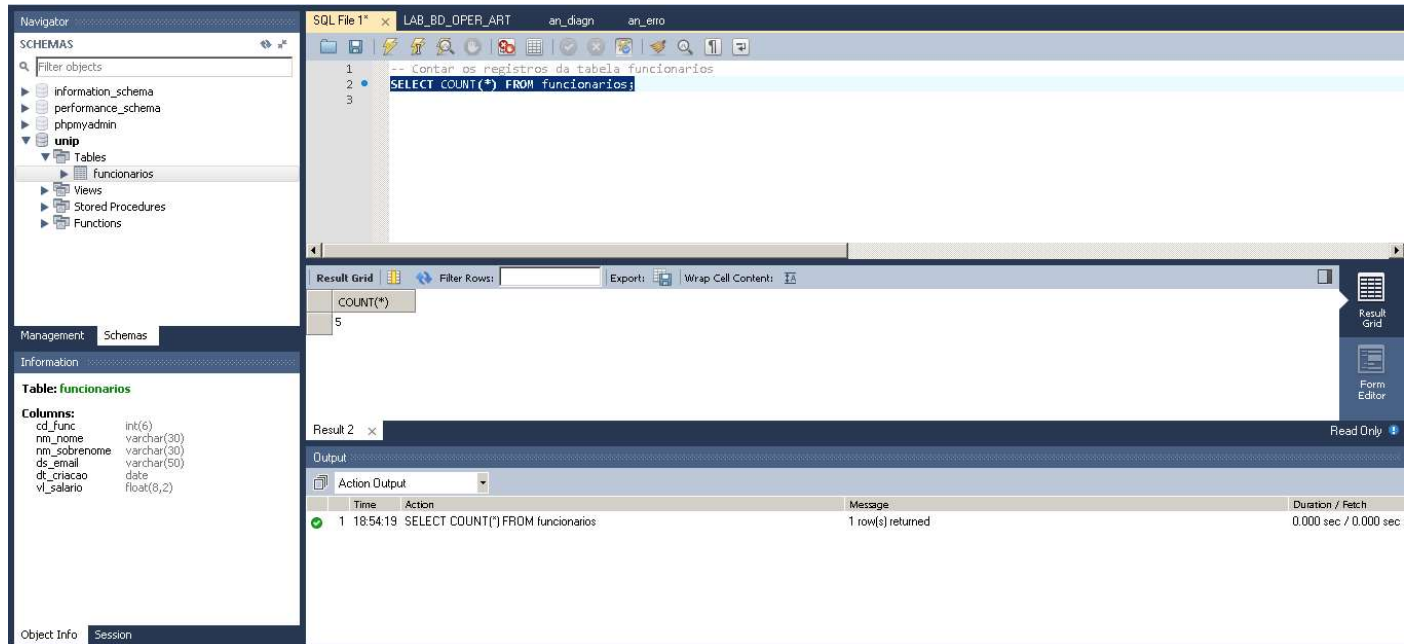
Time	Action	Message	Duration / Fetch
1 11:26:14	DESCRIBE funcionarios	6 row(s) returned	0.016 sec / 0.000 s

Efetuating queries in MySQL using Aggregation Functions

We will start counting the records in our table, without establishing filters in the WHERE clause:

```
-- Contar os registros da tabela funcionarios
```

```
SELECT COUNT(*) FROM funcionarios;
```

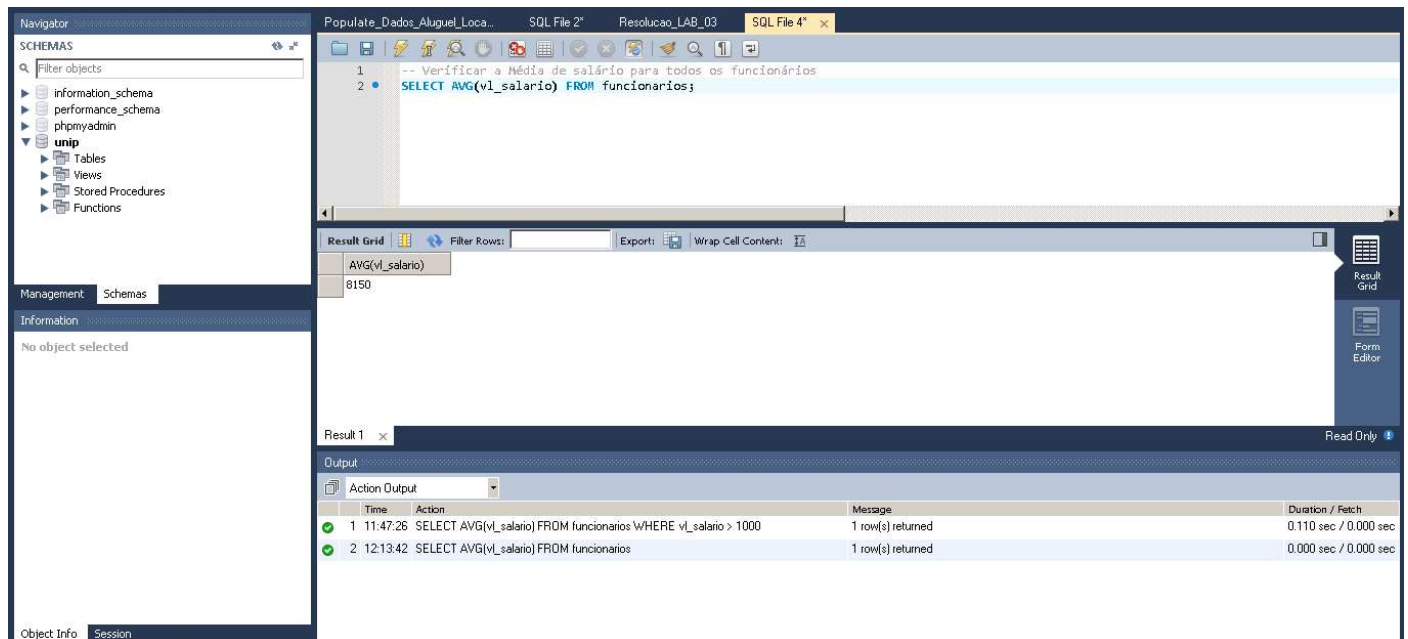


The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including the 'funcionarios' table. The main editor window shows the SQL query: `-- Contar os registros da tabela funcionarios` followed by `SELECT COUNT(*) FROM funcionarios;`. Below the editor, the 'Result Grid' shows a single row with the value '5' under the column 'COUNT(*)'. The 'Output' pane at the bottom shows the execution details: '1 18:54:19 SELECT COUNT(*) FROM funcionarios' with a message '1 row(s) returned' and a duration of '0.000 sec / 0.000 sec'.

Using the AVG Aggregation Function, we will consult the average of the salaries of the employees:

```
-- Verificar a Média de salário para todos os funcionários
```

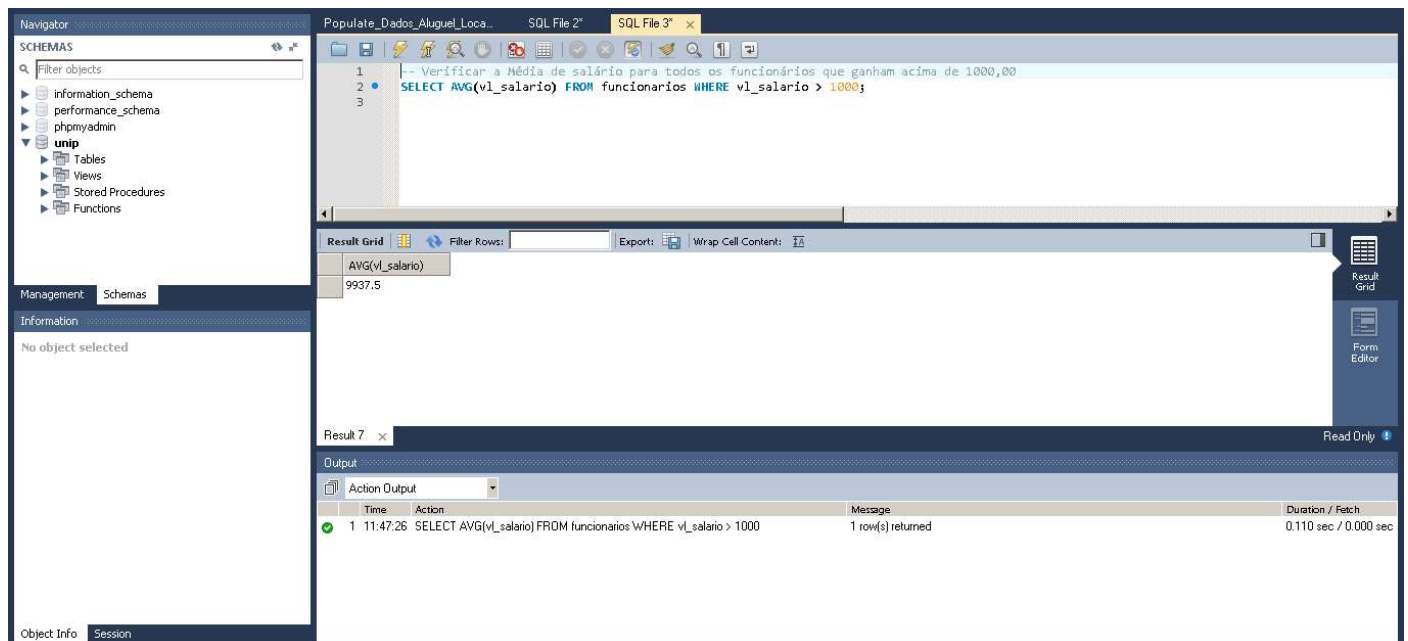
```
SELECT AVG(vl_salario) FROM funcionarios;
```



The screenshot shows the MySQL Workbench interface. The main editor window shows the SQL query: `-- Verificar a Média de salário para todos os funcionários` followed by `SELECT AVG(vl_salario) FROM funcionarios;`. Below the editor, the 'Result Grid' shows a single row with the value '8150' under the column 'AVG(vl_salario)'. The 'Output' pane at the bottom shows the execution details: '1 11:47:26 SELECT AVG(vl_salario) FROM funcionarios WHERE vl_salario > 1000' with a message '1 row(s) returned' and a duration of '0.110 sec / 0.000 sec', and '2 12:13:42 SELECT AVG(vl_salario) FROM funcionarios' with a message '1 row(s) returned' and a duration of '0.000 sec / 0.000 sec'.

Neste comando vamos aplicar a mesma função, porém estabelecendo um filtro na clausula WHERE:

```
-- Verificar a Média de salário para todos os funcionários que ganham acima de 1000,00
SELECT AVG(vl_salario) FROM funcionarios
WHERE vl_salario > 1000;
```



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'unip' selected. The main pane shows a SQL query editor with the following query:

```
-- Verificar a Média de salário para todos os funcionários que ganham acima de 1000,00
SELECT AVG(vl_salario) FROM funcionarios WHERE vl_salario > 1000;
```

The 'Result Grid' shows the result of the query:

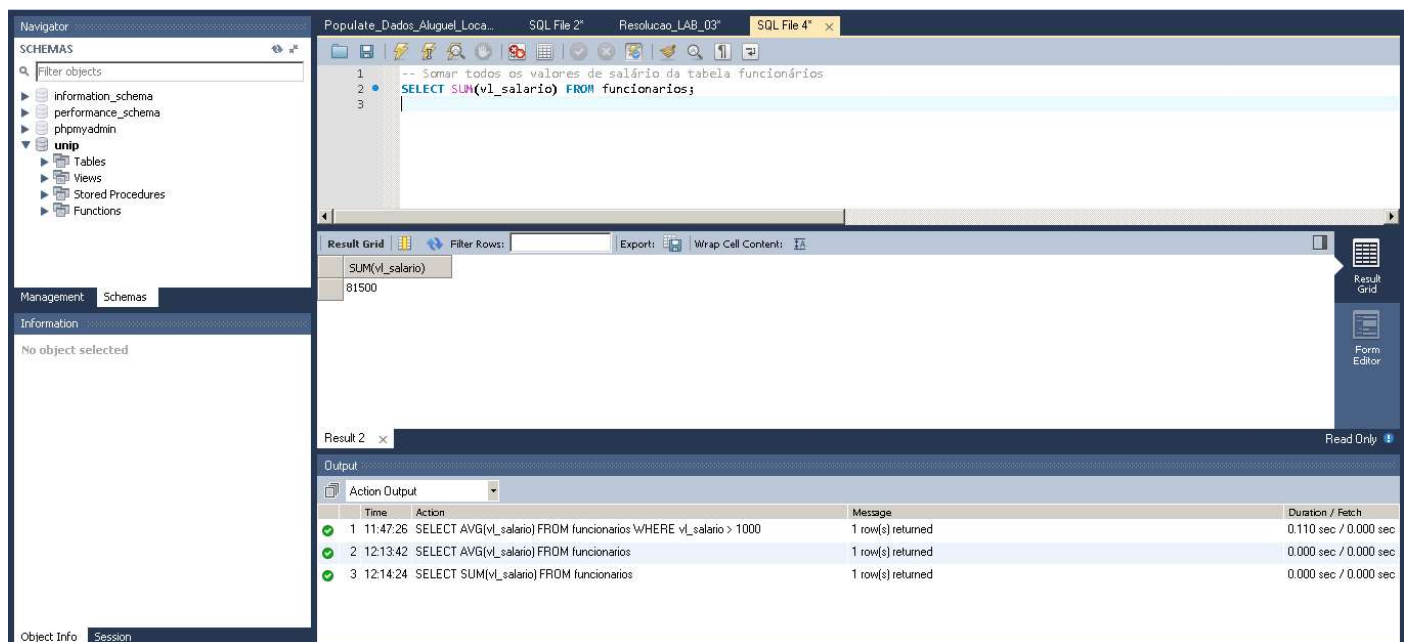
AVG(vl_salario)
9937.5

The 'Output' pane shows the 'Action Output' table with the following data:

Time	Action	Message	Duration / Fetch
1 11:47:26	SELECT AVG(vl_salario) FROM funcionarios WHERE vl_salario > 1000	1 row(s) returned	0.110 sec / 0.000 sec

Podemos somar os valores de salário dos registros presentes na tabela; para isso vamos utilizar a função SUM:

```
-- Somar todos os valores de salário da tabela funcionários
SELECT SUM(vl_salario) FROM funcionarios;
```



The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'unip' selected. The main pane shows a SQL query editor with the following query:

```
-- Somar todos os valores de salário da tabela funcionários
SELECT SUM(vl_salario) FROM funcionarios;
```

The 'Result Grid' shows the result of the query:

SUM(vl_salario)
81500

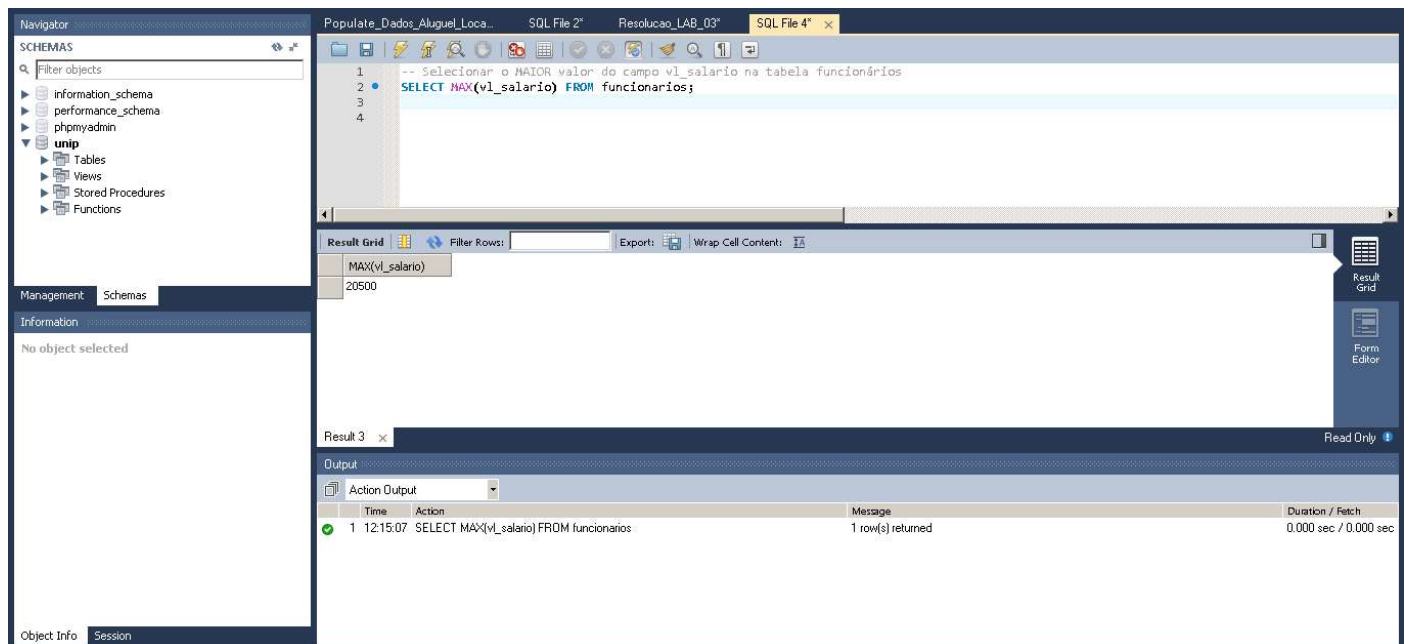
The 'Output' pane shows the 'Action Output' table with the following data:

Time	Action	Message	Duration / Fetch
1 11:47:26	SELECT AVG(vl_salario) FROM funcionarios WHERE vl_salario > 1000	1 row(s) returned	0.110 sec / 0.000 sec
2 12:13:42	SELECT AVG(vl_salario) FROM funcionarios	1 row(s) returned	0.000 sec / 0.000 sec
3 12:14:24	SELECT SUM(vl_salario) FROM funcionarios	1 row(s) returned	0.000 sec / 0.000 sec

Podemos selecionar o maior valor da coluna vl_salario utilizando a função MAX:

-- Selecionar o MAIOR valor do campo vl_salario na tabela funcionarios

```
SELECT MAX(vl_salario) FROM funcionarios;
```

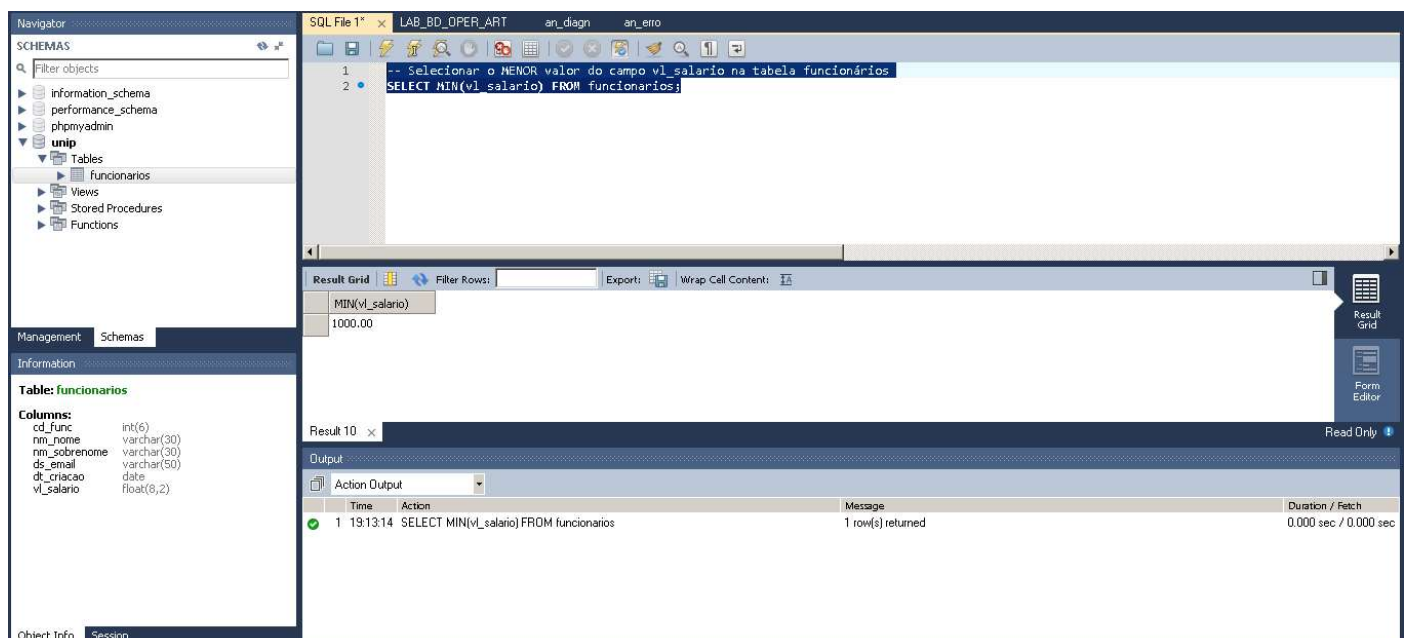


The screenshot shows the SQL Studio interface. The SQL editor contains the query: `-- Selecionar o MAIOR valor do campo vl_salario na tabela funcionarios` followed by `SELECT MAX(vl_salario) FROM funcionarios;`. The Result Grid shows a single row with the value 20500. The Output pane shows the execution message: "1 12:15:07 SELECT MAX(vl_salario) FROM funcionarios 1 row(s) returned 0.000 sec / 0.000 sec".

Também podemos selecionar o MENOR valor da coluna vl_salario utilizando a função MIN:

-- Selecionar o MENOR valor do campo vl_salario na tabela funcionarios

```
SELECT MIN(vl_salario) FROM funcionarios;
```



The screenshot shows the SQL Studio interface. The SQL editor contains the query: `-- Selecionar o MENOR valor do campo vl_salario na tabela funcionarios` followed by `SELECT MIN(vl_salario) FROM funcionarios;`. The Result Grid shows a single row with the value 1000.00. The Output pane shows the execution message: "1 19:13:14 SELECT MIN(vl_salario) FROM funcionarios 1 row(s) returned 0.000 sec / 0.000 sec". The Information pane shows the structure of the 'funcionarios' table.

Column	Type
cd_func	int(6)
nm_nome	varchar(30)
nm_sobrenome	varchar(30)
ds_email	varchar(50)
dt_criacao	date
vl_salario	float(8,2)

Exercícios

- 1- Efetue uma consulta na tabela funcionarios que retorne o MAIOR valor de salário para os registros que possuem código MAIOR que 2.
- 2- Efetue uma consulta na tabela funcionarios e verifique a MÉDIA dos salários para os funcionários que possuem nome IGUAL a 'Netuno' ou sobrenome IGUAL a 'Trump'.
- 3- Efetue uma consulta na tabela funcionarios e verifique a MÉDIA dos salários para os funcionários que possuem nome IGUAL a 'Astrogildo'.
- 4- Efetue uma consulta na tabela funcionarios e verifique a quantidade de funcionários que possuem salário MENOR que a MÉDIA de todos.
- 5- Efetue uma consulta na tabela funcionarios e selecione NOME, SOBRENOME e SALARIO dos funcionários que possuem valor de salário IGUAL ou MENOR ao menor valor cadastrado.
- 6- Efetue uma consulta na tabela funcionarios e selecione o registro do funcionário com MENOR salário contratado (dt_criacao) no ano de 2020.
- 7- Efetue uma consulta na tabela funcionarios e verifique a SOMA dos salários para os funcionários contratados (dt_criacao) no ano de 2023 com valor de salário MENOR que a MÉDIA dos salários dos funcionários contratados em 2023.