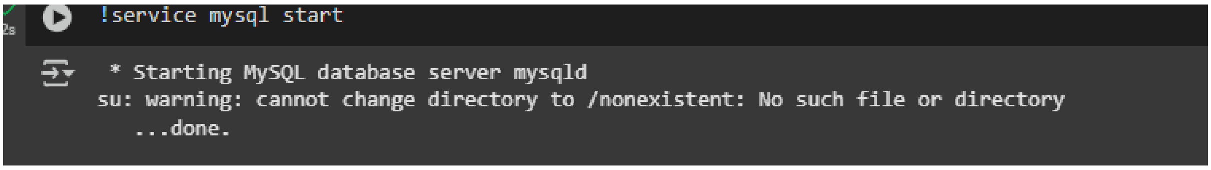
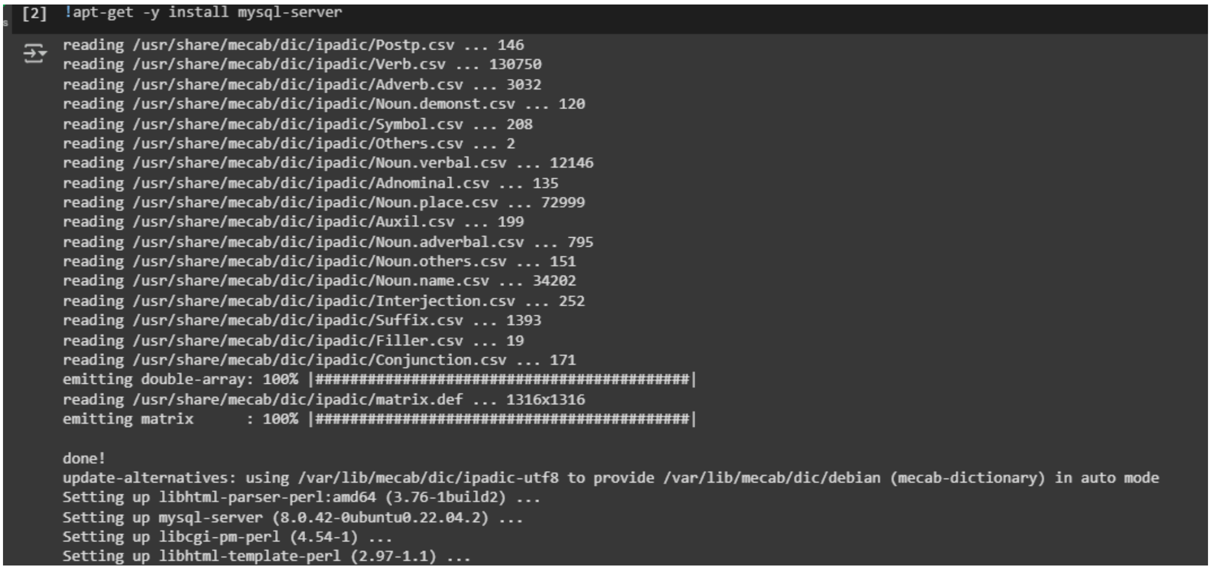
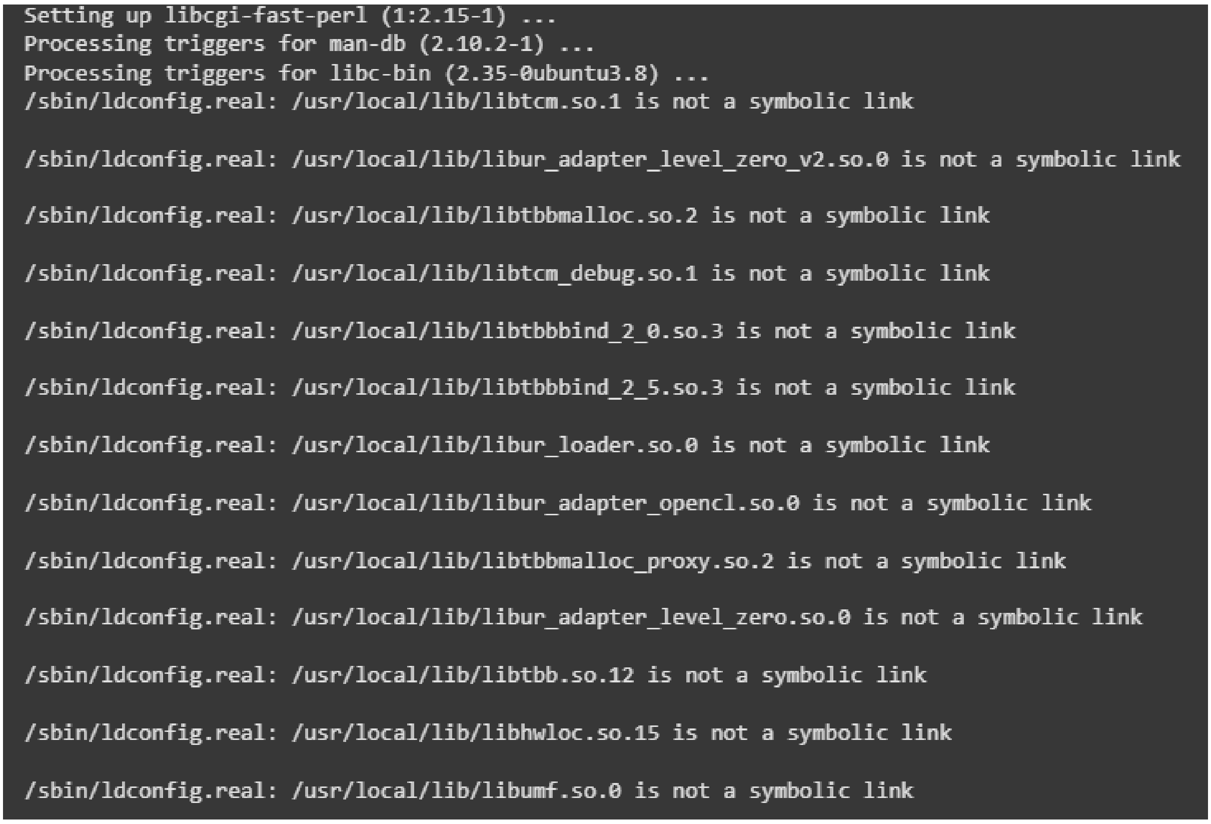


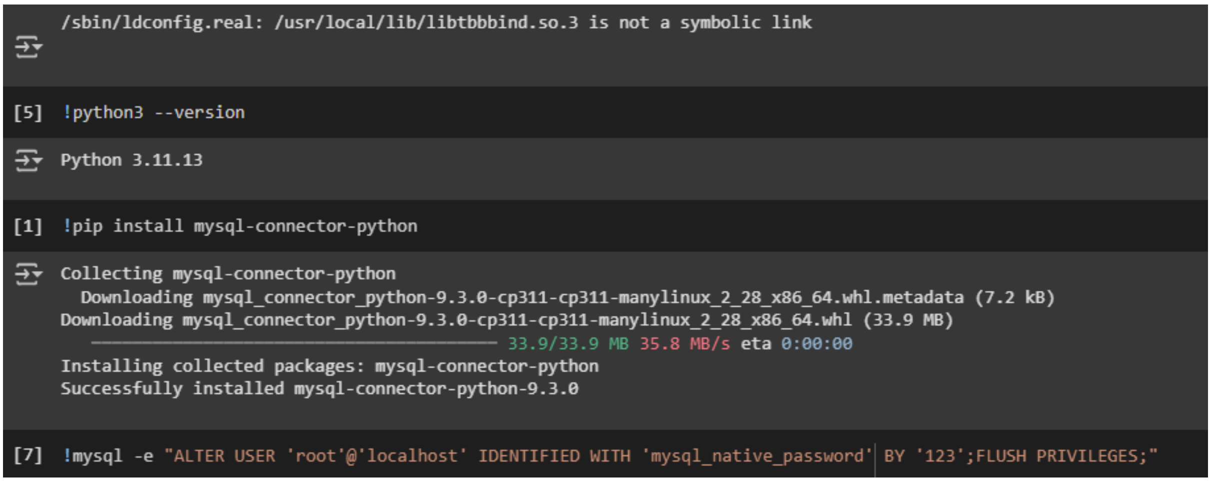
```
!service mysql start

 * Starting MySQL database server mysqld
su: warning: cannot change directory to /nonexistent: No such file or directory
   ...done.
```



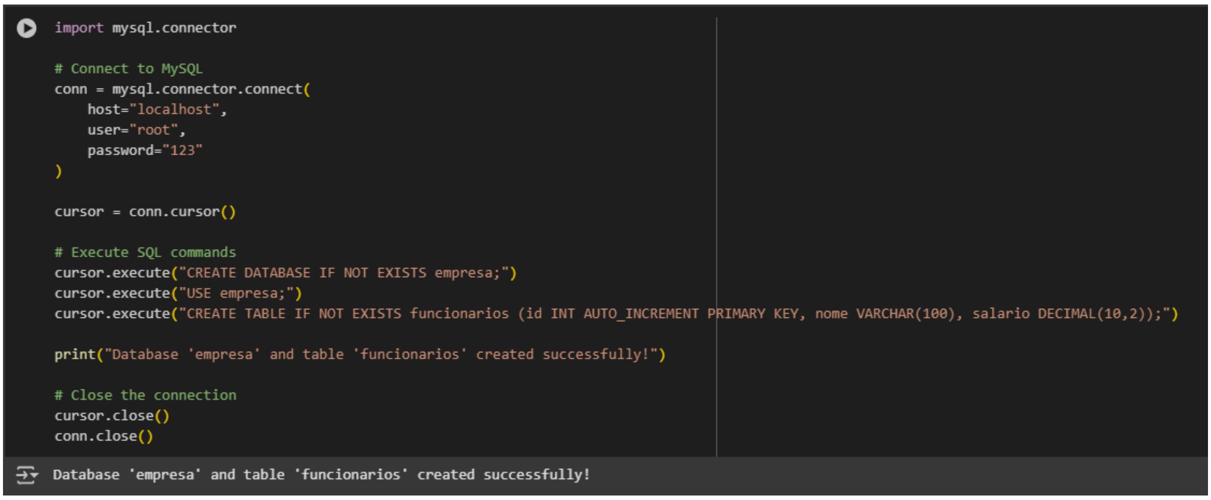
```
[2]  !apt-get -y install mysql-server

reading /usr/share/mecab/dic/ipadic/Postp.csv ... 146
reading /usr/share/mecab/dic/ipadic/Verb.csv ... 130750
reading /usr/share/mecab/dic/ipadic/Adverb.csv ... 3032
reading /usr/share/mecab/dic/ipadic/Noun.demonst.csv ... 120
reading /usr/share/mecab/dic/ipadic/Symbol.csv ... 208
reading /usr/share/mecab/dic/ipadic/Others.csv ... 2
reading /usr/share/mecab/dic/ipadic/Noun.verbal.csv ... 12146
reading /usr/share/mecab/dic/ipadic/Adnominal.csv ... 135
reading /usr/share/mecab/dic/ipadic/Noun.place.csv ... 72999
reading /usr/share/mecab/dic/ipadic/Auxil.csv ... 199
reading /usr/share/mecab/dic/ipadic/Noun.adverbal.csv ... 795
reading /usr/share/mecab/dic/ipadic/Noun.others.csv ... 151
reading /usr/share/mecab/dic/ipadic/Noun.name.csv ... 34202
reading /usr/share/mecab/dic/ipadic/Interjection.csv ... 252
reading /usr/share/mecab/dic/ipadic/Suffix.csv ... 1393
reading /usr/share/mecab/dic/ipadic/Filler.csv ... 19
reading /usr/share/mecab/dic/ipadic/Conjunction.csv ... 171
emitting double-array: 100% |###########################################|
reading /usr/share/mecab/dic/ipadic/matrix.def ... 1316x1316
emitting matrix      : 100% |###########################################|

done!
update-alternatives: using /var/lib/mecab/dic/ipadic-utf8 to provide /var/lib/mecab/dic/debian (mecab-dictionary) in auto mode
Setting up libhtml-parser-perl:amd64 (3.76-1build2) ...
Setting up mysql-server (8.0.42-0ubuntu0.22.04.2) ...
Setting up libcgi-pm-perl (4.54-1) ...
Setting up libhtml-template-perl (2.97-1.1) ...
```



```
Setting up libcgi-fast-perl (1:2.15-1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero_v2.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link
```

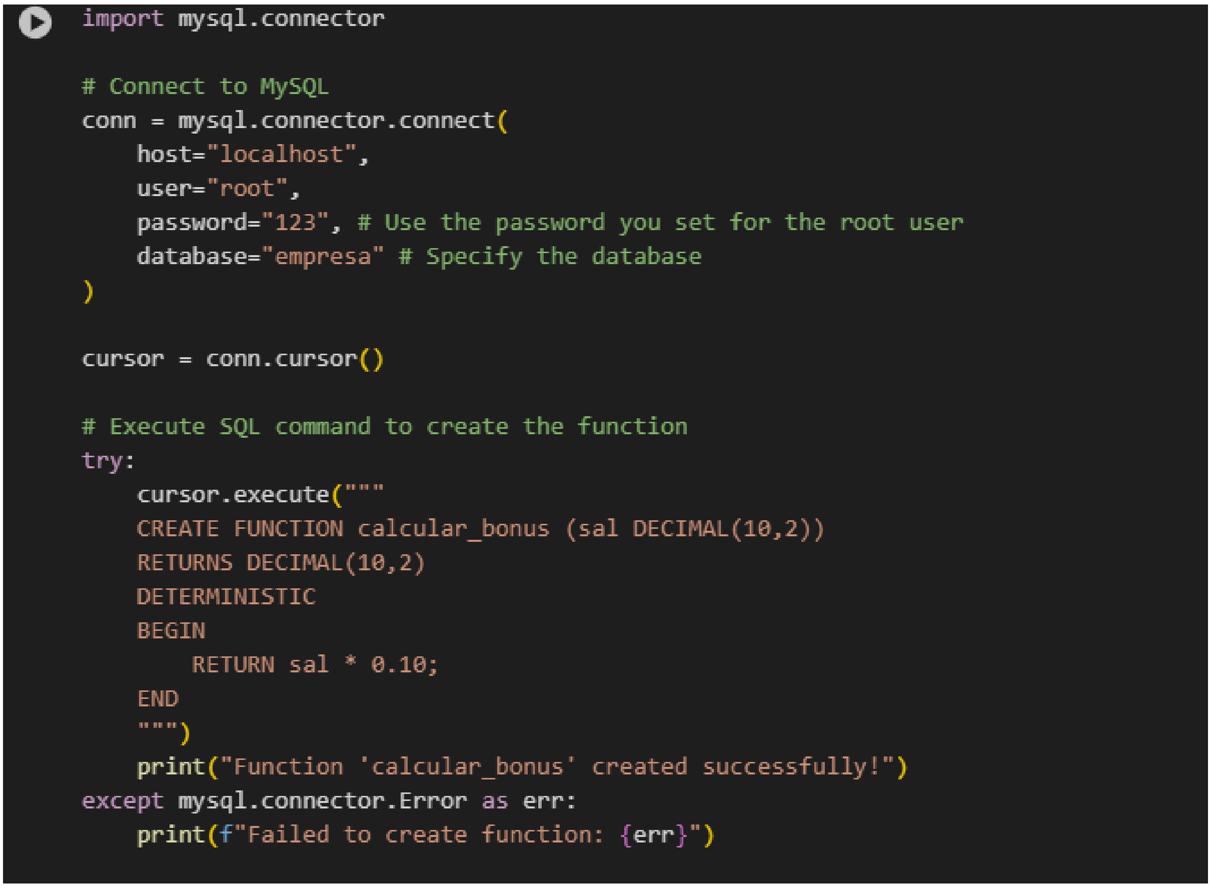


```
/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

[5] !python3 --version

Python 3.11.13

[1] !pip install mysql-connector-python

Collecting mysql-connector-python
  Downloading mysql_connector_python-9.3.0-cp311-cp311-manylinux_2_28_x86_64.whl.metadata (7.2 kB)
Downloading mysql_connector_python-9.3.0-cp311-cp311-manylinux_2_28_x86_64.whl (33.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 33.9/33.9 MB 35.8 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.3.0

[7] !mysql -e "ALTER USER 'root'@'localhost' IDENTIFIED WITH 'mysql_native_password' BY '123';FLUSH PRIVILEGES;"
```

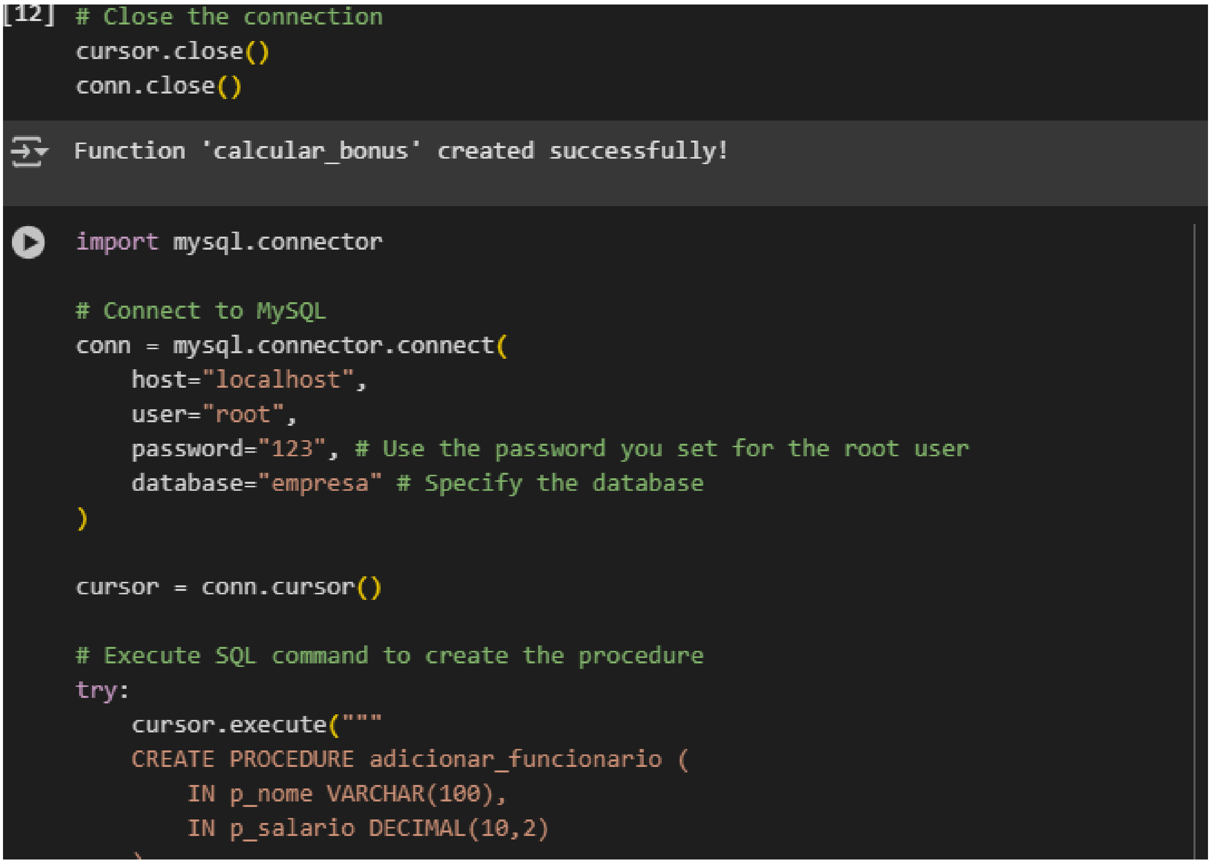


```
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123"
)

cursor = conn.cursor()

# Execute SQL commands
cursor.execute("CREATE DATABASE IF NOT EXISTS empresa;")
cursor.execute("USE empresa;")
cursor.execute("CREATE TABLE IF NOT EXISTS funcionarios (id INT AUTO_INCREMENT PRIMARY KEY, nome VARCHAR(100), salario DECIMAL(10,2));")

print("Database 'empresa' and table 'funcionarios' created successfully!")

# Close the connection
cursor.close()
conn.close()
```

```
Database 'empresa' and table 'funcionarios' created successfully!
```



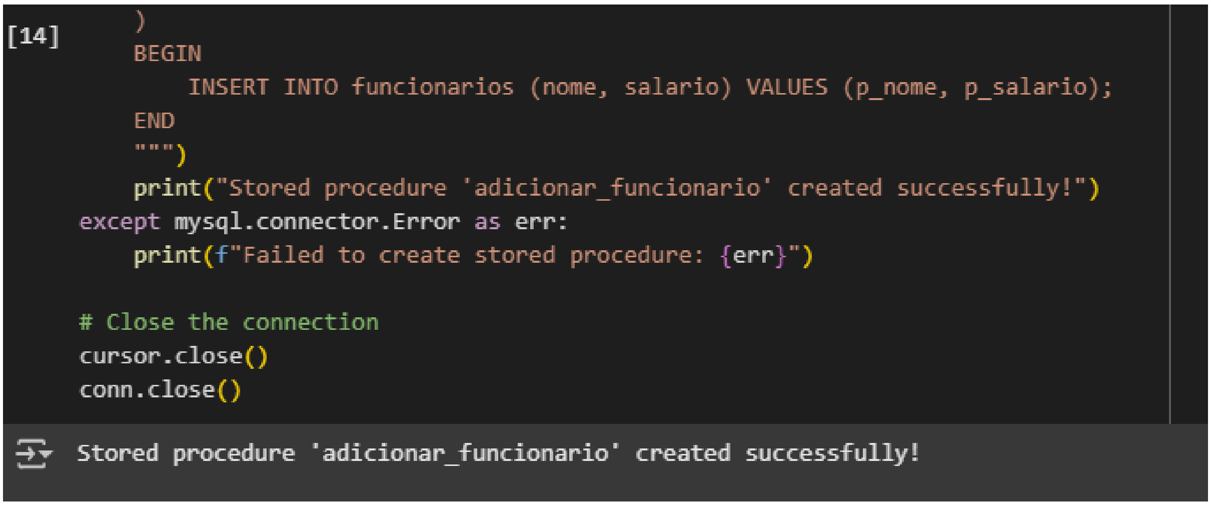
```python
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123", # Use the password you set for the root user
    database="empresa" # Specify the database
)

cursor = conn.cursor()

# Execute SQL command to create the function
try:
    cursor.execute("""
    CREATE FUNCTION calcular_bonus (sal DECIMAL(10,2))
    RETURNS DECIMAL(10,2)
    DETERMINISTIC
    BEGIN
        RETURN sal * 0.10;
    END
    """)
    print("Function 'calcular_bonus' created successfully!")
except mysql.connector.Error as err:
    print(f"Failed to create function: {err}")
```



```python
[12] # Close the connection
     cursor.close()
     conn.close()
```

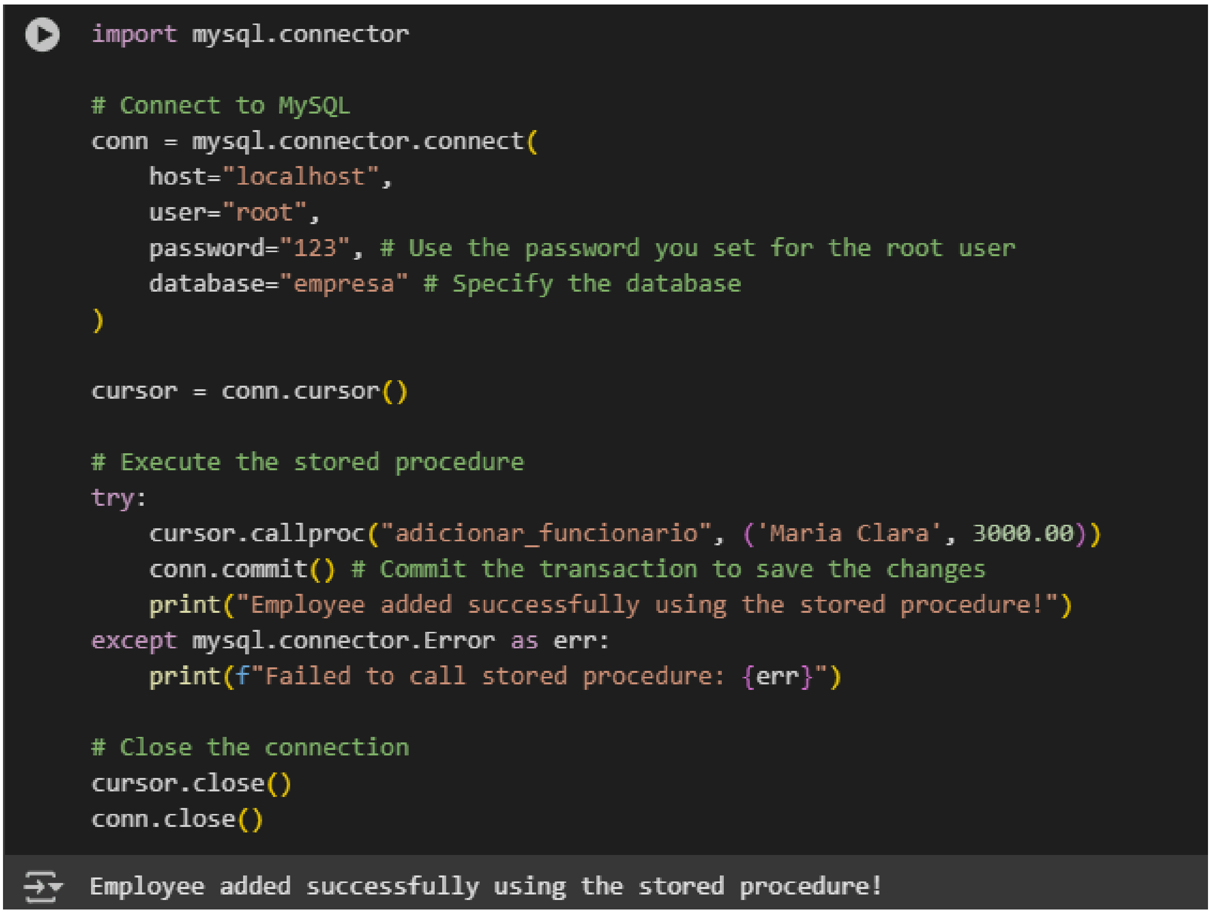
```
Function 'calcular_bonus' created successfully!
```

```python
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123", # Use the password you set for the root user
    database="empresa" # Specify the database
)

cursor = conn.cursor()

# Execute SQL command to create the procedure
try:
    cursor.execute("""
    CREATE PROCEDURE adicionar_funcionario (
        IN p_nome VARCHAR(100),
        IN p_salario DECIMAL(10,2)
```



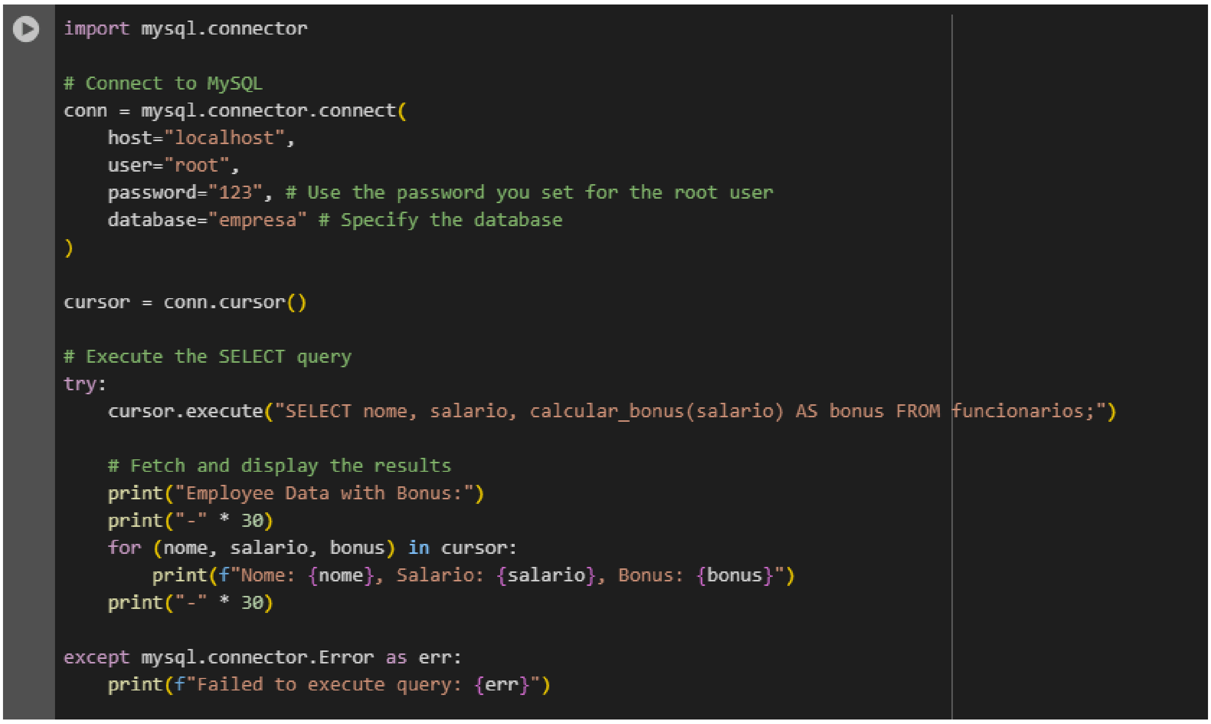
```
    )
    BEGIN
        INSERT INTO funcionarios (nome, salario) VALUES (p_nome, p_salario);
    END
    """)
    print("Stored procedure 'adicionar_funcionario' created successfully!")
except mysql.connector.Error as err:
    print(f"Failed to create stored procedure: {err}")

# Close the connection
cursor.close()
conn.close()
```

```
Stored procedure 'adicionar_funcionario' created successfully!
```

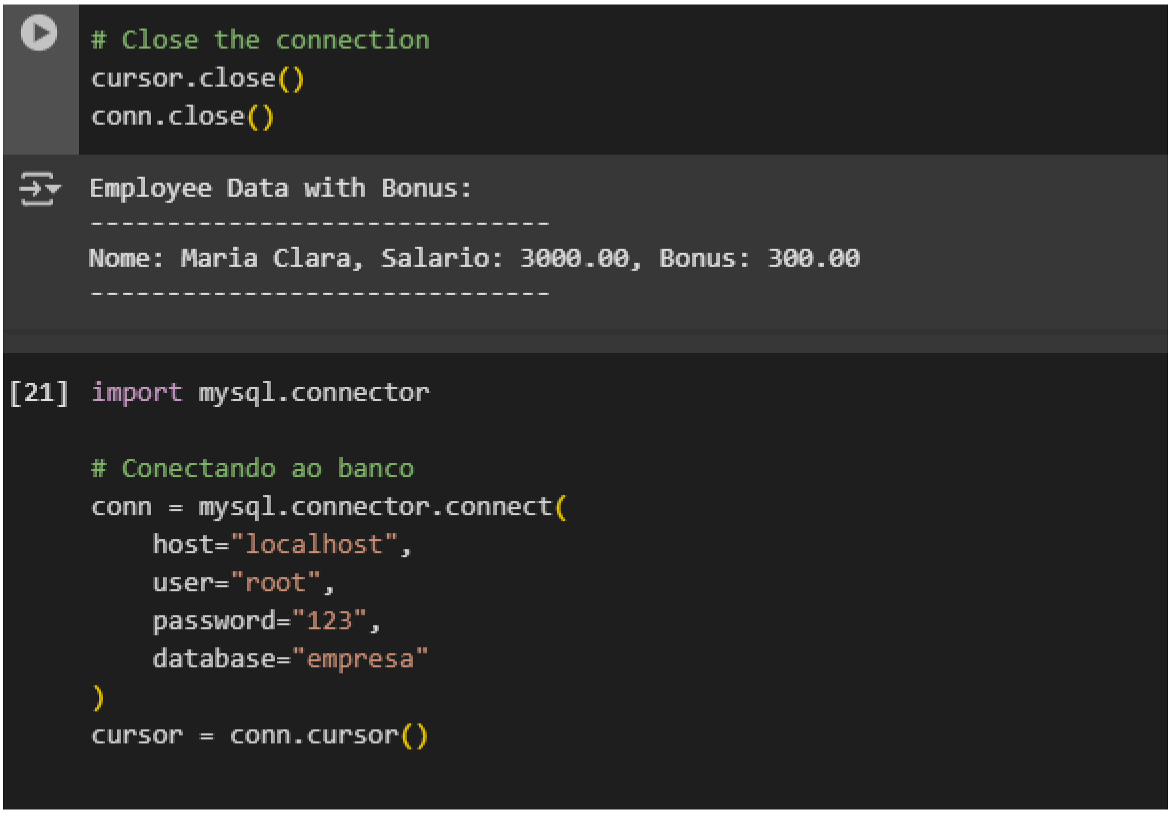


```
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123", # Use the password you set for the root user
    database="empresa" # Specify the database
)

cursor = conn.cursor()

# Execute the stored procedure
try:
    cursor.callproc("adicionar_funcionario", ('Maria Clara', 3000.00))
    conn.commit() # Commit the transaction to save the changes
    print("Employee added successfully using the stored procedure!")
except mysql.connector.Error as err:
    print(f"Failed to call stored procedure: {err}")

# Close the connection
cursor.close()
conn.close()
```

```
Employee added successfully using the stored procedure!
```



```python
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123", # Use the password you set for the root user
    database="empresa" # Specify the database
)

cursor = conn.cursor()

# Execute the SELECT query
try:
    cursor.execute("SELECT nome, salario, calcular_bonus(salario) AS bonus FROM funcionarios;")

    # Fetch and display the results
    print("Employee Data with Bonus:")
    print("-" * 30)
    for (nome, salario, bonus) in cursor:
        print(f"Nome: {nome}, Salario: {salario}, Bonus: {bonus}")
    print("-" * 30)

except mysql.connector.Error as err:
    print(f"Failed to execute query: {err}")
```

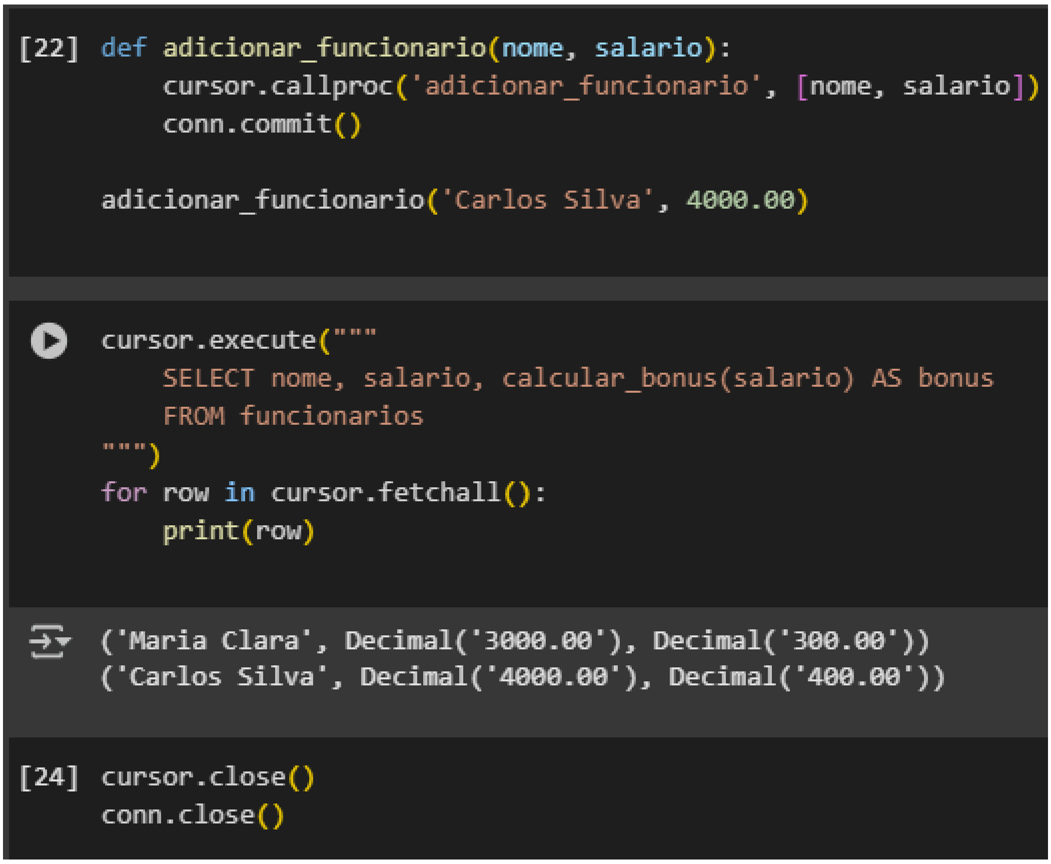


```python
# Close the connection
cursor.close()
conn.close()
```

```
Employee Data with Bonus:
------------------------------
Nome: Maria Clara, Salario: 3000.00, Bonus: 300.00
------------------------------
```

[21]

```python
import mysql.connector

# Conectando ao banco
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="123",
    database="empresa"
)
cursor = conn.cursor()
```



```
def adicionar_funcionario(nome, salario):
    cursor.callproc('adicionar_funcionario', [nome, salario])
    conn.commit()

adicionar_funcionario('Carlos Silva', 4000.00)
```

```
cursor.execute("""
    SELECT nome, salario, calcular_bonus(salario) AS bonus
    FROM funcionarios
""")
for row in cursor.fetchall():
    print(row)
```

```
('Maria Clara', Decimal('3000.00'), Decimal('300.00'))
('Carlos Silva', Decimal('4000.00'), Decimal('400.00'))
```

```
cursor.close()
conn.close()
```