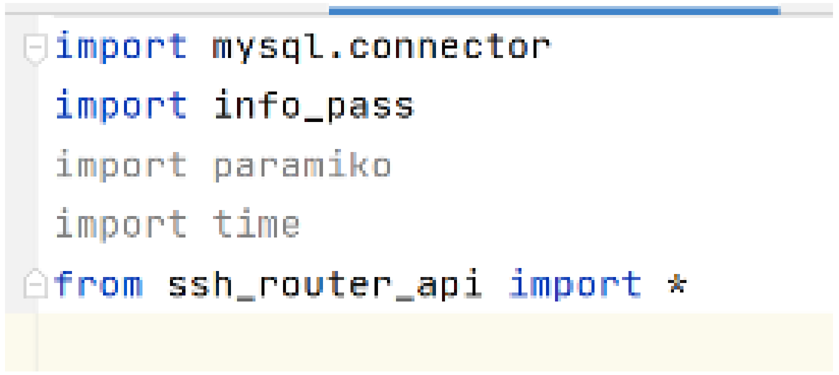
Python and MySQL

Network Application

Install mysql-connector **or** mysql-connector-python

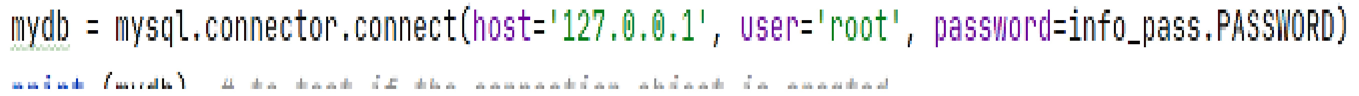
- After installing mysql-connector or mysql-connector-python, import mysql.connector (ensure that Pycharm also installs one of these connectors that connects to MySQL)



```
import mysql.connector
import info_pass
import paramiko
import time
from ssh_router_api import *
```

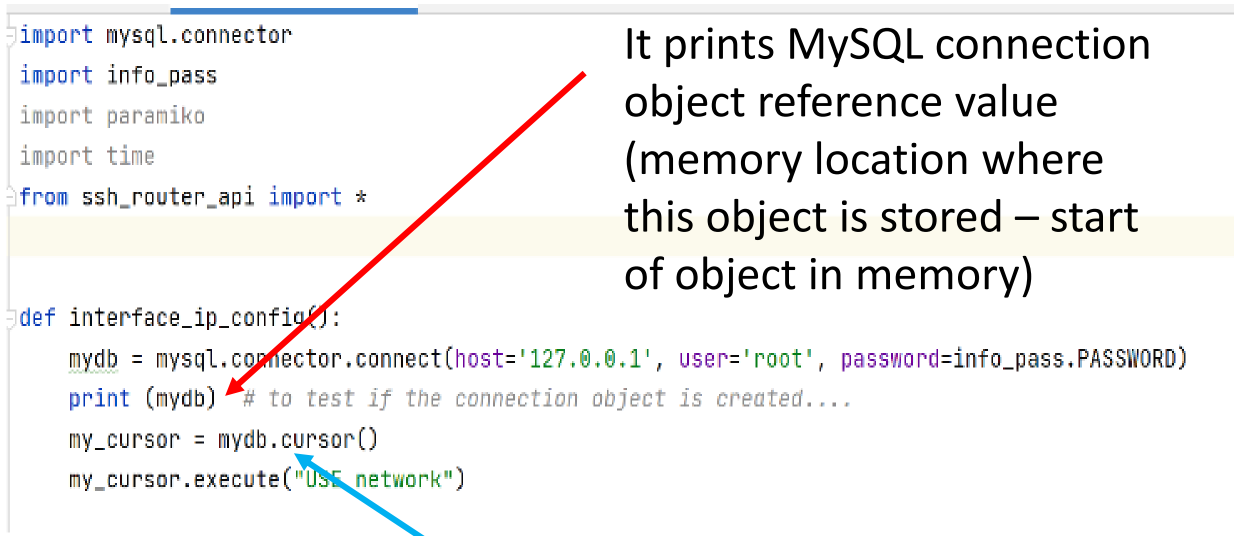
Connecting to MySQL database

info_pass is also imported and contains password



```
mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
print (mydb)  # to test if the connection object is created
```

Accessing database rows

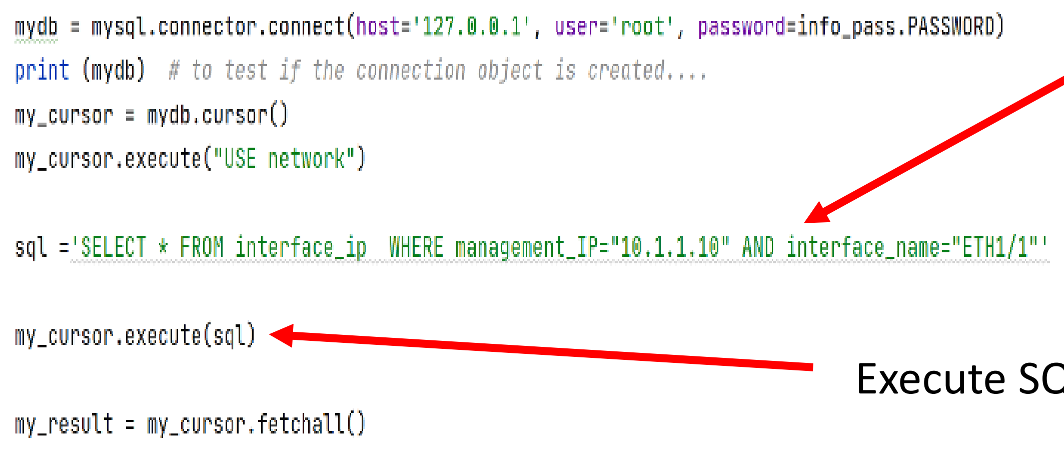


```
import mysql.connector
import info_pass
import paramiko
import time
from ssh_router_api import *


def interface_ip_config():
    mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
    print (mydb)  # to test if the connection object is created....
    my_cursor = mydb.cursor()
    my_cursor.execute("USE network")
```

It prints MySQL connection object reference value (memory location where this object is stored – start of object in memory)

Cursor is Database Management System approach to access one or more rows of data generated by executing SQL query

SQL query execution and data retrieval



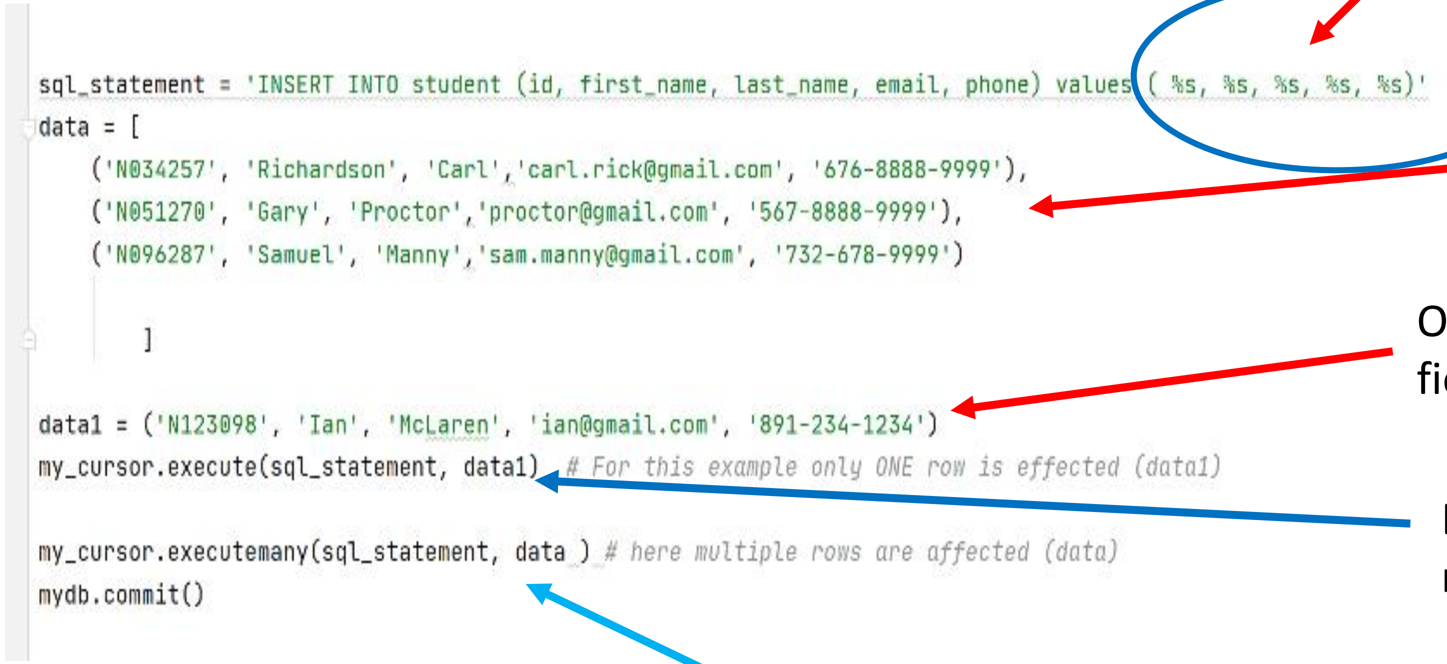
```
mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
print (mydb)  # to test if the connection object is created....
my_cursor = mydb.cursor()
my_cursor.execute("USE network")

sql ='SELECT * FROM interface_ip  WHERE management_IP="10.1.1.10" AND interface_name="ETH1/1"'

my_cursor.execute(sql)

my_result = my_cursor.fetchall()
```

SQL query

Execute SQL query

Fetch data from cursor to variable to display

Insert one or multiple row of data



```
sql_statement = 'INSERT INTO student (id, first_name, last_name, email, phone) values ( %s, %s, %s, %s, %s)'
data = [
    ('N034257', 'Richardson', 'Carl','carl.rick@gmail.com', '676-8888-9999'),
    ('N051270', 'Gary', 'Proctor','proctor@gmail.com', '567-8888-9999'),
    ('N096287', 'Samuel', 'Manny','sam.manny@gmail.com', '732-678-9999')

        ]

data1 = ('N123098', 'Ian', 'McLaren', 'ian@gmail.com', '891-234-1234')
my_cursor.execute(sql_statement, data1)  # For this example only ONE row is effected (data1)

my_cursor.executemany(sql_statement, data ) # here multiple rows are affected (data)
mydb.commit()
```

Place holder for different fields

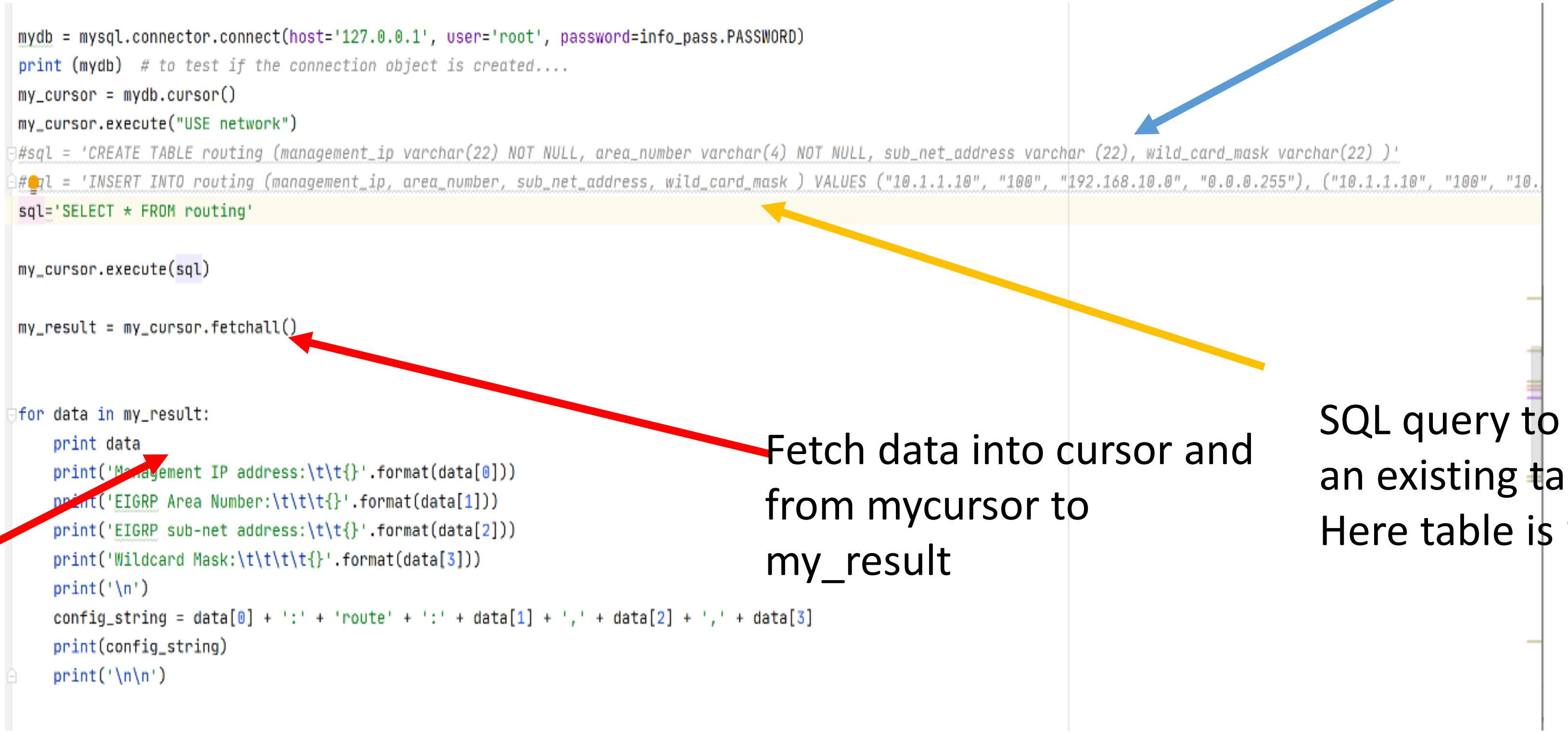
List of tuples – multiple rows of data

One row of data with multiple fields

Execute SQL query for one row only

Execute query for multiple rows
NOTE: **‘executemany’**

Create table, Insert data and fetch information;



```
mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
print (mydb)  # to test if the connection object is created....
my_cursor = mydb.cursor()
my_cursor.execute("USE network")
#sql = 'CREATE TABLE routing (management_ip varchar(22) NOT NULL, area_number varchar(4) NOT NULL, sub_net_address varchar (22), wild_card_mask varchar(22) )'
#sql = 'INSERT INTO routing (management_ip, area_number, sub_net_address, wild_card_mask ) VALUES ("10.1.1.10", "100", "192.168.10.0", "0.0.0.255"), ("10.1.1.10", "100", "10.
sql='SELECT * FROM routing'

my_cursor.execute(sql)

my_result = my_cursor.fetchall()


for data in my_result:
    print data
    print('Management IP address:\t\t{}'.format(data[0]))
    print('EIGRP Area Number:\t\t\t{}'.format(data[1]))
    print('EIGRP sub-net address:\t\t{}'.format(data[2]))
    print('Wildcard Mask:\t\t\t\t{}'.format(data[3]))
    print('\n')
    config_string = data[0] + ':' + 'route' + ':' + data[1] + ',' + data[2] + ',' + data[3]
    print(config_string)
    print('\n\n')
```

SQL query to create table – ‘routing’

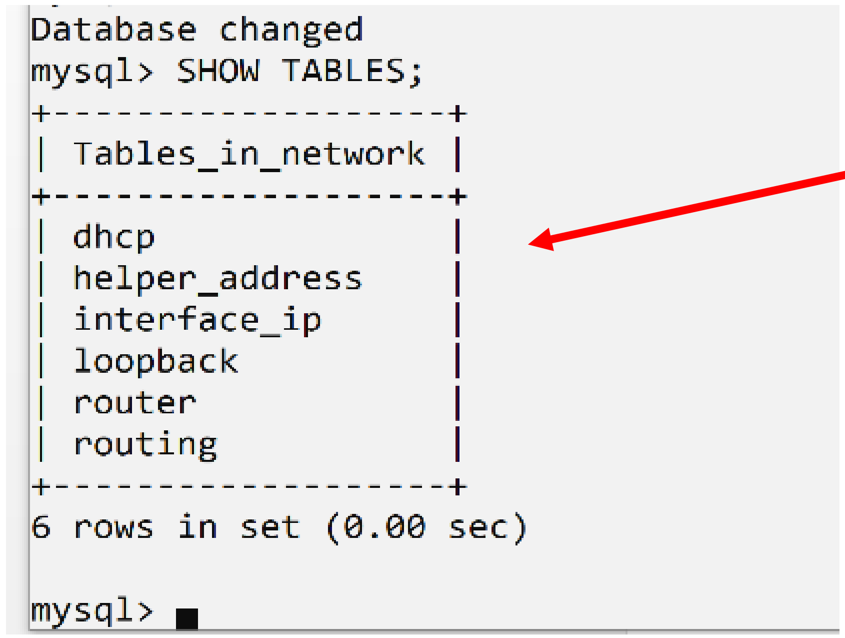
SQL query to insert data in an existing table;
Here table is ‘**routing**’

Fetch data into cursor and from mycursor to my_result

Display result from fetchall

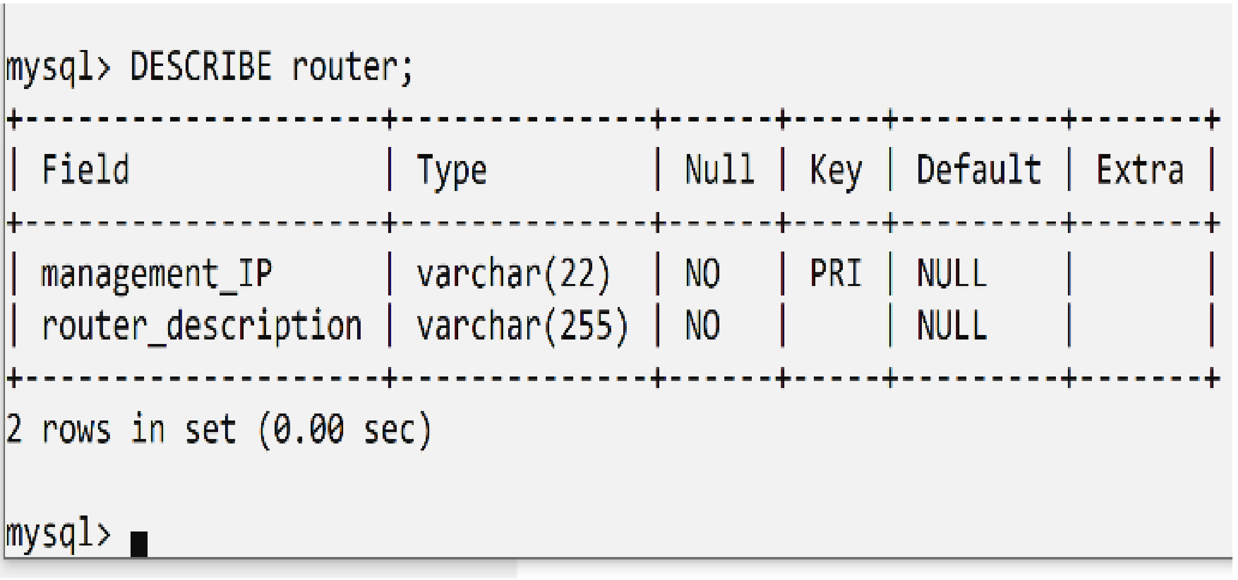
Network Automation – Example to configure organizational network

Tables used in database ‘network’

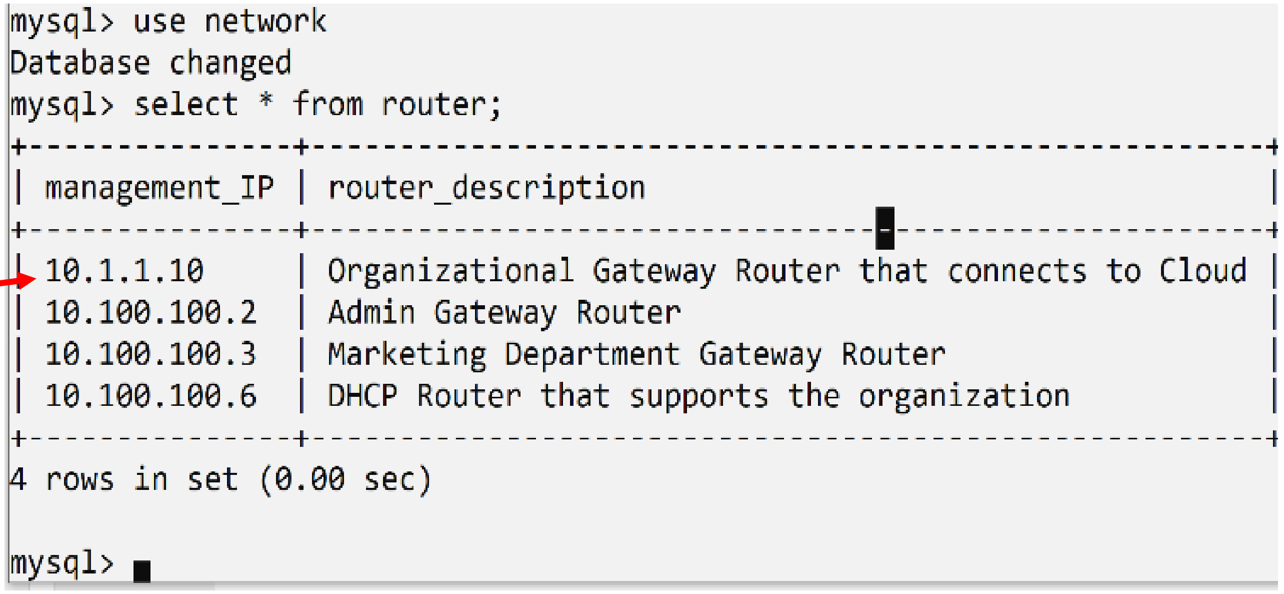


```
Database changed
mysql> SHOW TABLES;
+--------------------+
| Tables_in_network |
+--------------------+
| dhcp               |
| helper_address     |
| interface_ip       |
| loopback           |
| router             |
| routing            |
+--------------------+
6 rows in set (0.00 sec)

mysql> 
```

Main Table (router)– Management IP address order is important



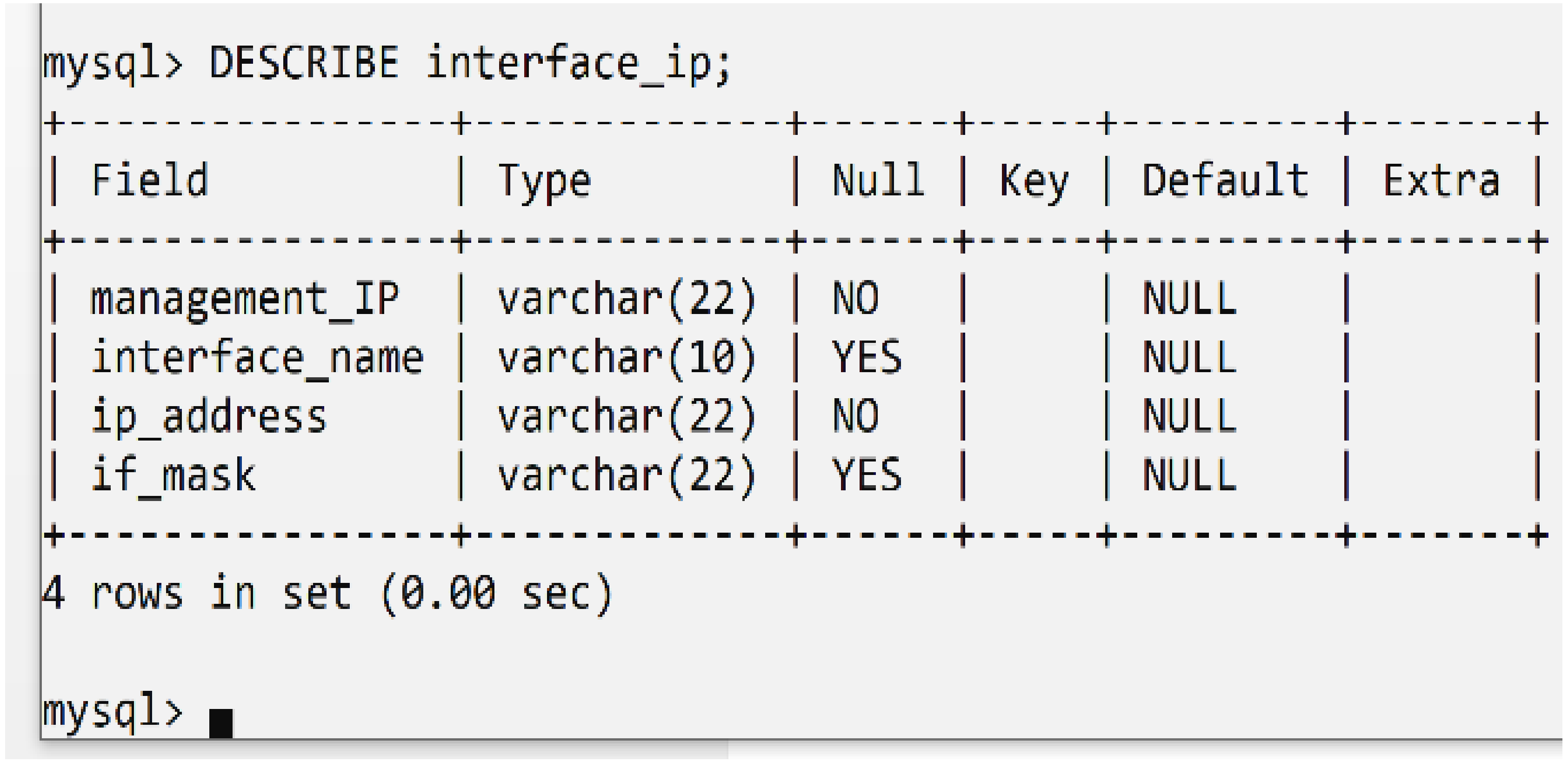
```
mysql> DESCRIBE router;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| management_IP      | varchar(22)  | NO   | PRI | NULL    |       |
| router_description | varchar(255) | NO   |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
2 rows in set (0.00 sec)

mysql>
```



```
mysql> use network
Database changed
mysql> select * from router;
+---------------+-----------------------------------------------------+
| management_IP | router_description                                  |
+---------------+-----------------------------------------------------+
| 10.1.1.10     | Organizational Gateway Router that connects to Cloud |
| 10.100.100.2  | Admin Gateway Router                                |
| 10.100.100.3  | Marketing Department Gateway Router                 |
| 10.100.100.6  | DHCP Router that supports the organization          |
+---------------+-----------------------------------------------------+
4 rows in set (0.00 sec)

mysql>
```

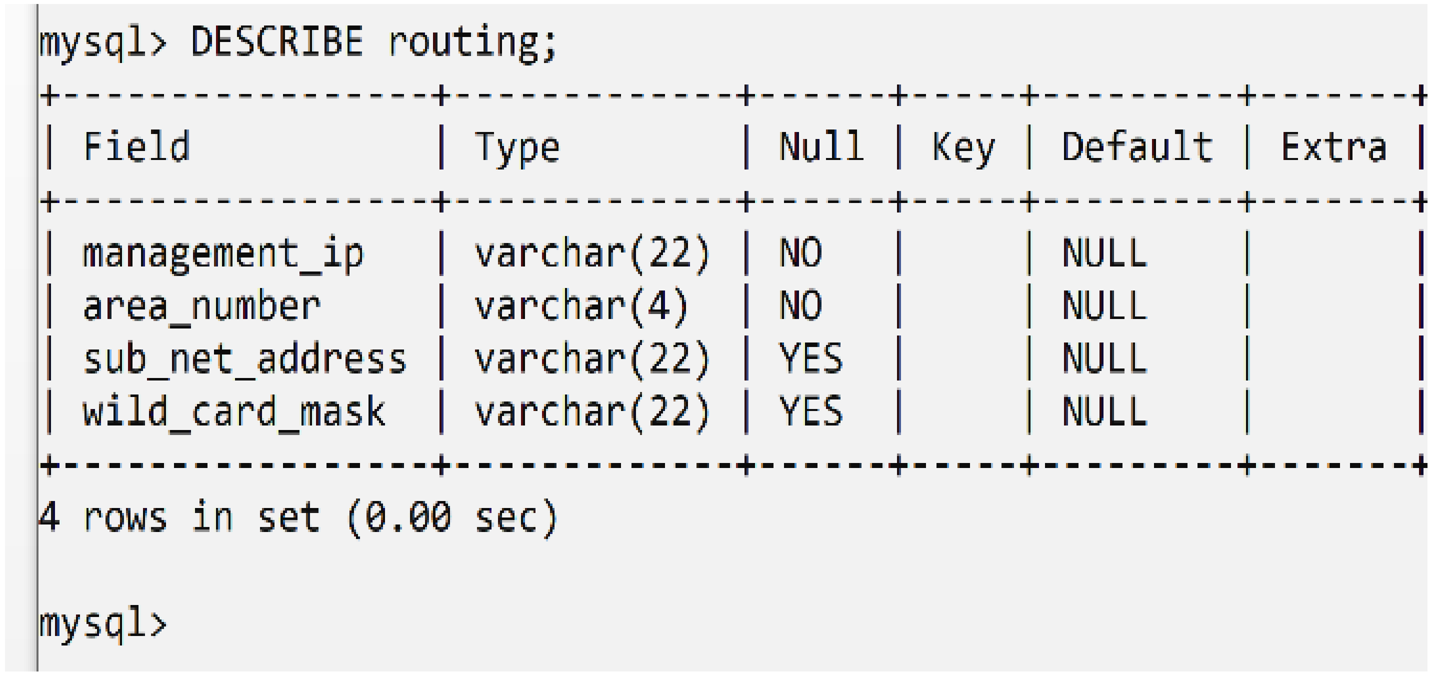
First – Organizational Gateway Router, connected to Cloud must be configured first

Assigning interface IP addresses – use table ‘interface_ip’



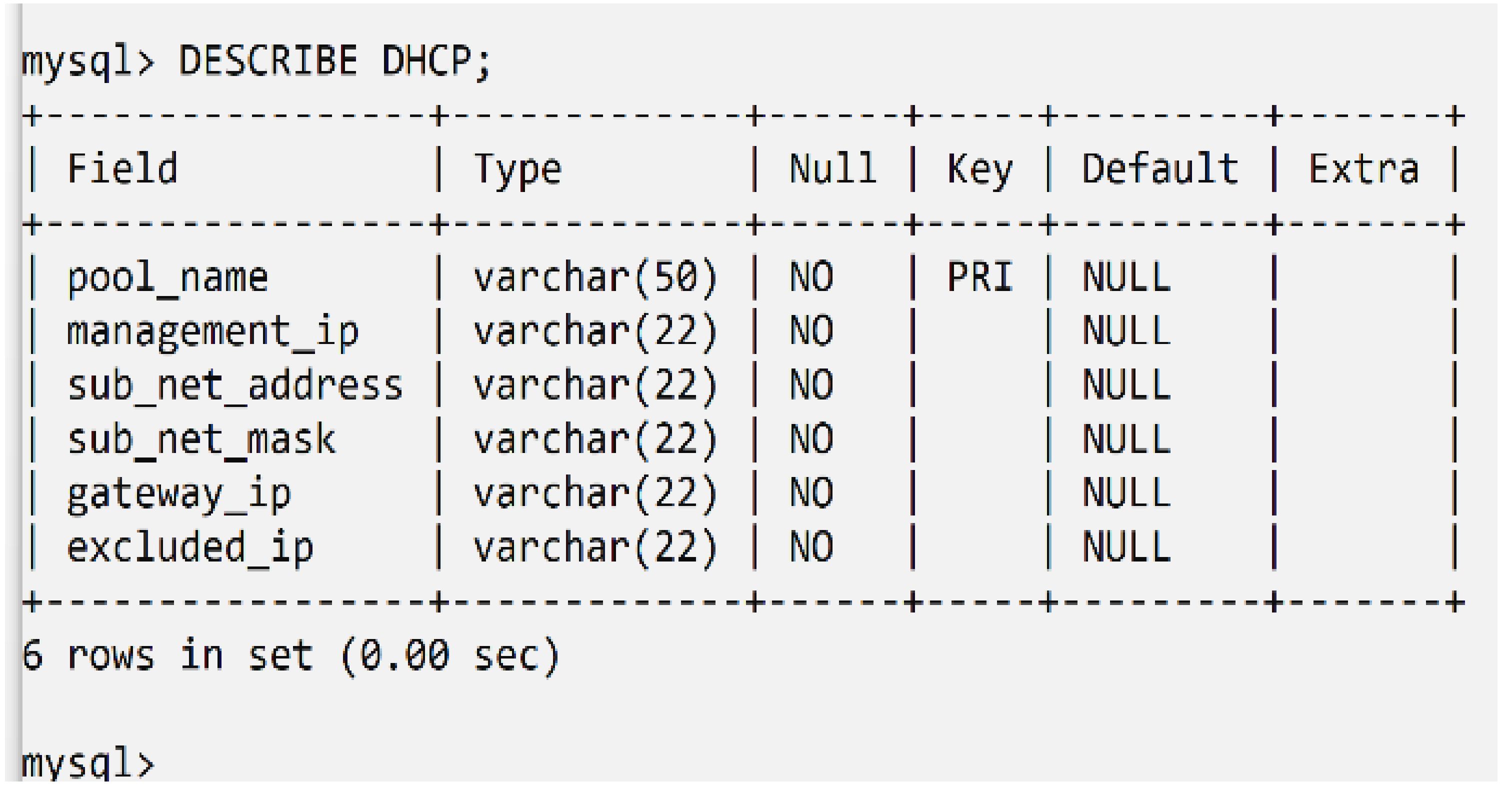
```
mysql> DESCRIBE interface_ip;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| management_IP   | varchar(22) | NO   |     | NULL    |       |
| interface_name  | varchar(10) | YES  |     | NULL    |       |
| ip_address      | varchar(22) | NO   |     | NULL    |       |
| if_mask         | varchar(22) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> 
```

Assign routing – use table ‘routing’



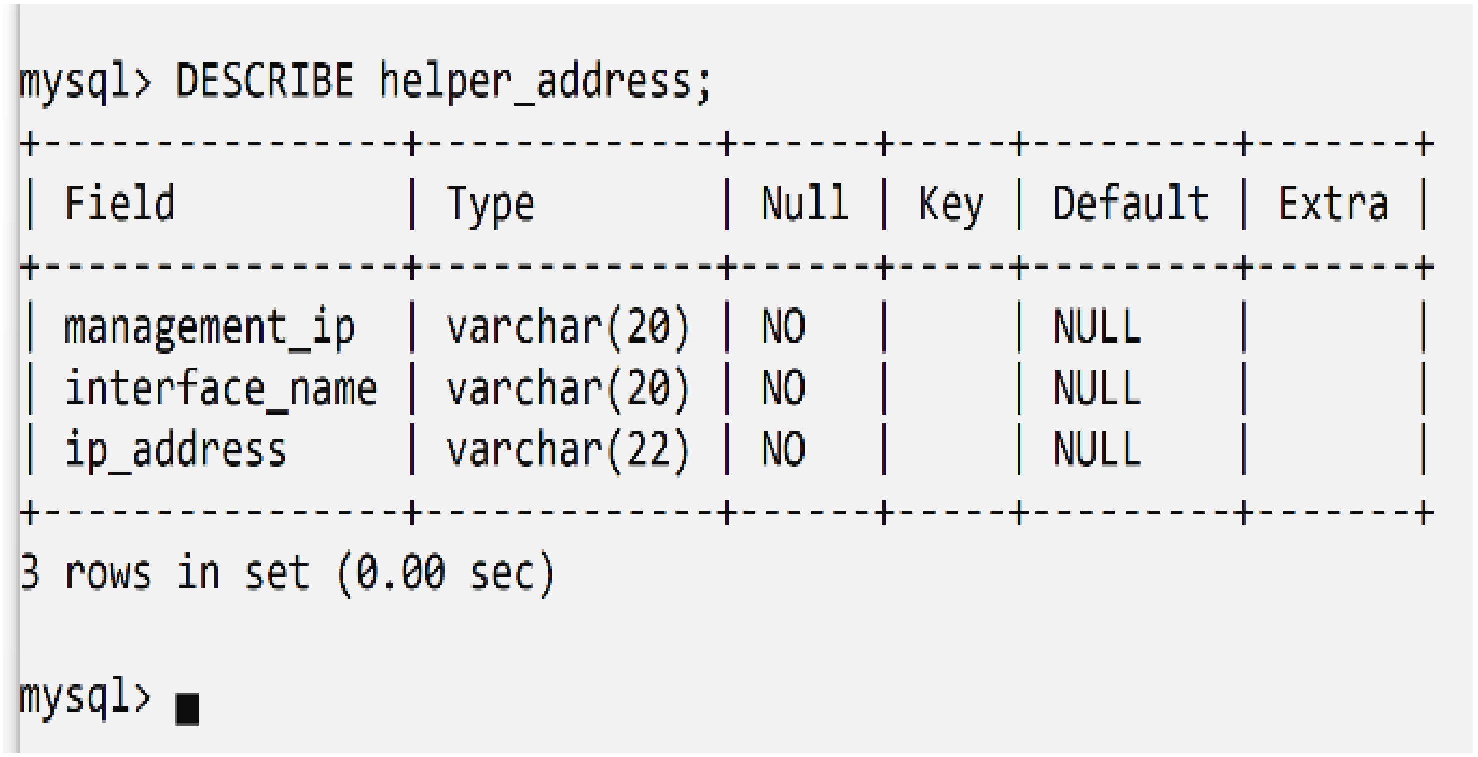
```
mysql> DESCRIBE routing;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| management_ip   | varchar(22) | NO   |     | NULL    |       |
| area_number     | varchar(4)  | NO   |     | NULL    |       |
| sub_net_address | varchar(22) | YES  |     | NULL    |       |
| wild_card_mask  | varchar(22) | YES  |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql>
```

DHCP configuration using ‘dhcp’ table



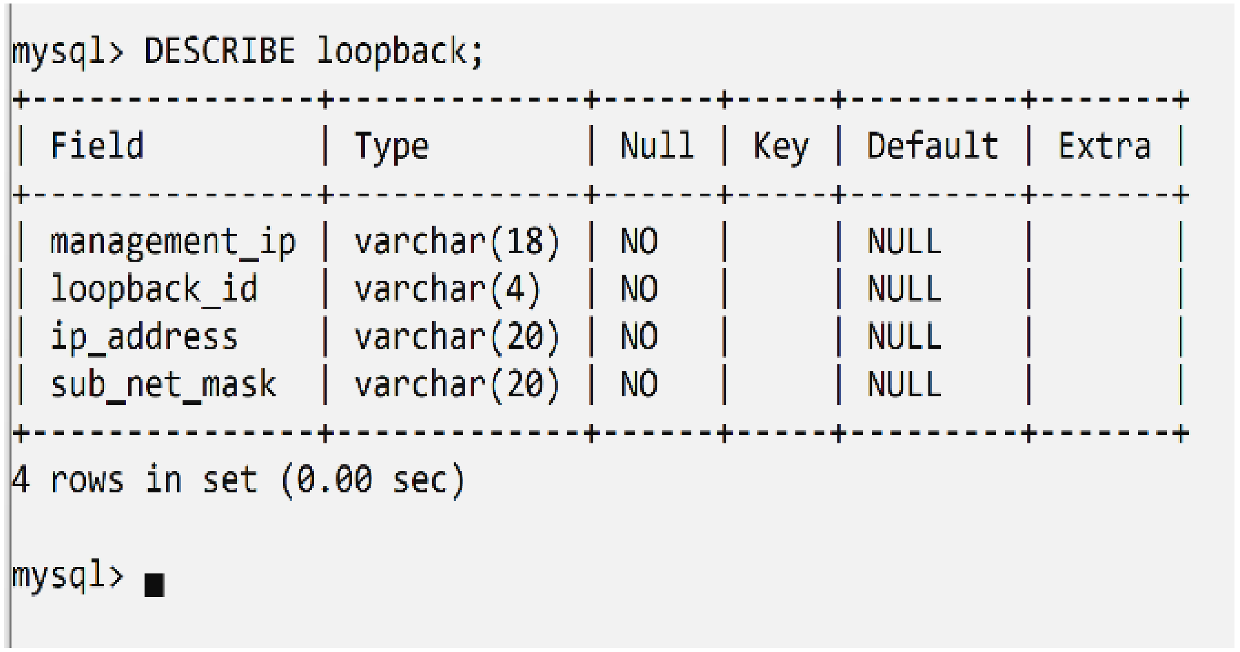
```
mysql> DESCRIBE DHCP;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| pool_name       | varchar(50) | NO   | PRI | NULL    |       |
| management_ip   | varchar(22) | NO   |     | NULL    |       |
| sub_net_address | varchar(22) | NO   |     | NULL    |       |
| sub_net_mask    | varchar(22) | NO   |     | NULL    |       |
| gateway_ip      | varchar(22) | NO   |     | NULL    |       |
| excluded_ip     | varchar(22) | NO   |     | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

Helper Address configuration – using table ‘helper_address’



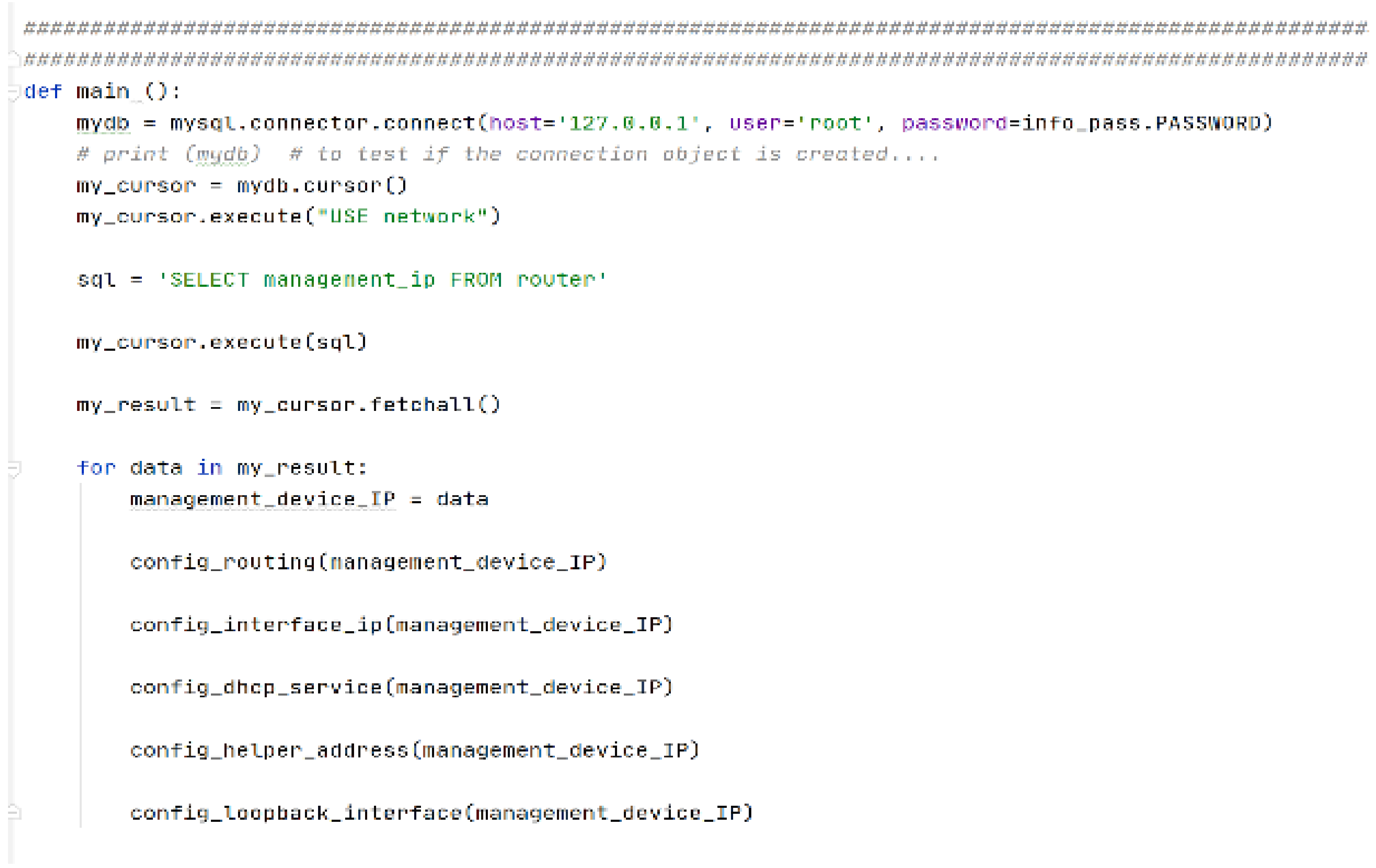
```
mysql> DESCRIBE helper_address;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| management_ip  | varchar(20) | NO   |     | NULL    |       |
| interface_name | varchar(20) | NO   |     | NULL    |       |
| ip_address     | varchar(22) | NO   |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> 
```

Loopback interface – create it using table ‘loopback’



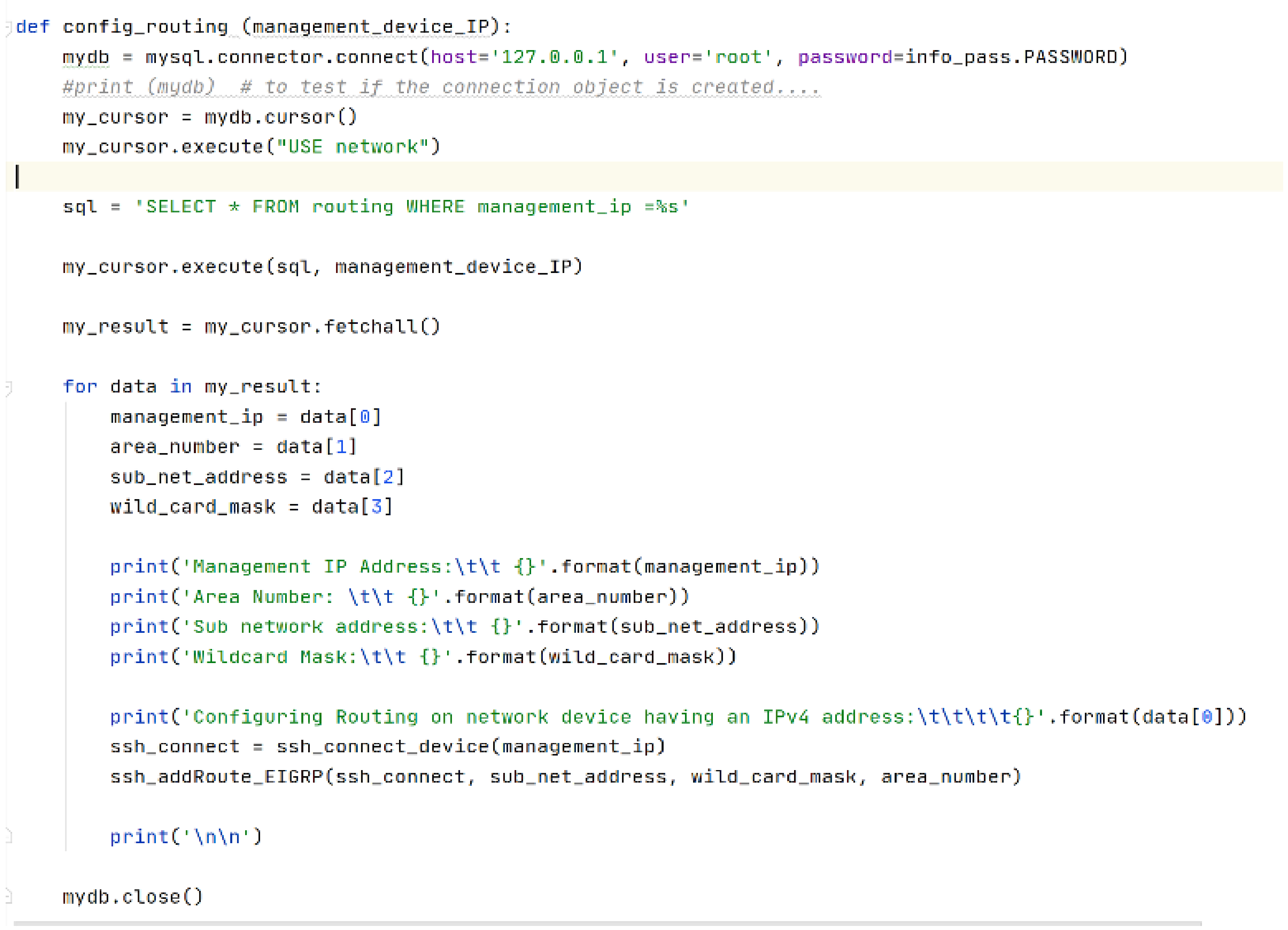
```
mysql> DESCRIBE loopback;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| management_ip | varchar(18) | NO   |     | NULL    |       |
| loopback_id   | varchar(4)  | NO   |     | NULL    |       |
| ip_address    | varchar(20) | NO   |     | NULL    |       |
| sub_net_mask  | varchar(20) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> 
```

Application starter – main function



```
######################################################################################################
######################################################################################################
def main ():
    mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
    # print (mydb)  # to test if the connection object is created....
    my_cursor = mydb.cursor()
    my_cursor.execute("USE network")

    sql = 'SELECT management_ip FROM router'

    my_cursor.execute(sql)

    my_result = my_cursor.fetchall()

    for data in my_result:
        management_device_IP = data

        config_routing(management_device_IP)

        config_interface_ip(management_device_IP)

        config_dhcp_service(management_device_IP)

        config_helper_address(management_device_IP)

        config_loopback_interface(management_device_IP)
```

Route configuration on devices



```
def config_routing (management_device_IP):
    mydb = mysql.connector.connect(host='127.0.0.1', user='root', password=info_pass.PASSWORD)
    #print (mydb)  # to test if the connection object is created....
    my_cursor = mydb.cursor()
    my_cursor.execute("USE network")

    sql = 'SELECT * FROM routing WHERE management_ip =%s'

    my_cursor.execute(sql, management_device_IP)

    my_result = my_cursor.fetchall()

    for data in my_result:
        management_ip = data[0]
        area_number = data[1]
        sub_net_address = data[2]
        wild_card_mask = data[3]

        print('Management IP Address:\t\t {}'.format(management_ip))
        print('Area Number: \t\t {}'.format(area_number))
        print('Sub network address:\t\t {}'.format(sub_net_address))
        print('Wildcard Mask:\t\t {}'.format(wild_card_mask))

        print('Configuring Routing on network device having an IPv4 address:\t\t\t\t{}'.format(data[0]))
        ssh_connect = ssh_connect_device(management_ip)
        ssh_addRoute_EIGRP(ssh_connect, sub_net_address, wild_card_mask, area_number)

        print('\n\n')

    mydb.close()
```