

In [3]: `!pip install mysql-connector-python`

Requirement already satisfied: mysql-connector-python in c:\users\bobbi\appdata\local\programs\python\python312\lib\site-packages (8.3.0)

[notice] A new release of pip is available: 23.3.2 -> 24.0

[notice] To update, run: python.exe -m pip install --upgrade pip

In [4]: `import mysql.connector as connector`

In [5]: `connection=connector.connect(user="root", password="peter", port=3306)`

In [6]: `cursor=connection.cursor()`

In [7]: `cursor.execute("USE little_lemon_db")`

In [8]: `# Create a stored procedure named GetMaxQuantity.  
cursor.execute("DROP PROCEDURE IF EXISTS GetMaxQuantity;")  
stored_procedure_query="""CREATE PROCEDURE GetMaxQuantity()  
BEGIN  
 SELECT MAX(quantity) as max_quantity  
 from orders;  
END  
"""`

In [9]: `cursor.execute(stored_procedure_query)`

In [13]: `cursor.callproc("GetMaxQuantity")`

Out[13]: `()`

In [14]: `results=next(cursor.stored_results())`

In [15]: `dataset=results.fetchall()`

In [16]: `for data in dataset:  
 print(data)`

`(5,)`

In [18]: `for column_id in cursor.stored_results():  
 cols=[column[0] for column in column_id.description]  
 print("Today's orders report:")  
 for result in dataset:  
 print("\t",cols[0],":",result[0])`

Today's orders report:  
max\_quantity : 5

In [19]: `# Create a stored procedure named ManageBooking.  
cursor.execute("DROP PROCEDURE IF EXISTS ManageBooking;")  
stored_procedure_query="""CREATE PROCEDURE ManageBooking()`

```
BEGIN
    SELECT * from bookings;
END
"""
```

```
In [20]: cursor.execute(stored_procedure_query)
```

```
In [23]: cursor.callproc("ManageBooking")
```

```
Out[23]: ()
```

```
In [24]: results=next(cursor.stored_results())
```

```
In [25]: dataset=results.fetchall()
```

```
In [28]: for column_id in cursor.stored_results():
          cols=[column[0] for column in column_id.description]

          for result in dataset:
              print("\t",cols[2],":",result[2], ' is already reserved ', 'on ', cols[3],":",

                    TableNumber : 12  is already reserved  on  BookingDate : 2022-12-11
                    TableNumber : 12  is already reserved  on  BookingDate : 2022-12-10
                    TableNumber : 15  is already reserved  on  BookingDate : 2022-12-08
                    TableNumber : 19  is already reserved  on  BookingDate : 2022-12-09
                    TableNumber : 5   is already reserved  on  BookingDate : 2022-12-11
                    TableNumber : 6   is already reserved  on  BookingDate : 2022-12-10
                    TableNumber : 8   is already reserved  on  BookingDate : 2022-12-08
                    TableNumber : 9   is already reserved  on  BookingDate : 2022-12-15
                    TableNumber : 11  is already reserved  on  BookingDate : 2022-12-14
                    TableNumber : 17  is already reserved  on  BookingDate : 2022-12-05
```

```
In [8]: # Create a stored procedure named AddBooking.
        cursor.execute("DROP PROCEDURE IF EXISTS AddBooking;")
        stored_procedure_query="""CREATE PROCEDURE AddBooking()
        BEGIN
            INSERT INTO Bookings(BookingID, CustomerID, Tablenumber, BookingDate) VALUES(99,

        END
        """
```

```
In [10]: cursor.execute(stored_procedure_query)
```

-----  
**MySQLInterfaceError** Traceback (most recent call last)  
File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\mysql\connector\connection\_cext.py:661, in **CMySQLConnection.cmd\_query**(self, query, raw, buffered, raw\_as\_string)

```
    660         query = query.encode("utf-8")
--> 661     self._cmysql.query(
    662         query,
    663         raw=raw,
    664         buffered=buffered,
    665         raw_as_string=raw_as_string,
    666         query_attrs=self.query_attrs,
    667     )
    668 except MySQLInterfaceError as err:
```

**MySQLInterfaceError**: PROCEDURE AddBooking already exists

The above exception was the direct cause of the following exception:

**ProgrammingError** Traceback (most recent call last)  
Cell In[10], line 1

```
----> 1 cursor.execute(stored_procedure_query)
```

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\mysql\connector\cursor\_cext.py:374, in **CMySQLCursor.execute**(self, operation, params, multi)

```
    369         raise ProgrammingError(
    370             "Not all parameters were used in the SQL statement"
    371         )
    373 try:
--> 374     result = self._cnx.cmd_query(
    375         stmt,
    376         raw=self._raw,
    377         buffered=self._buffered,
    378         raw_as_string=self._raw_as_string,
    379     )
    380 except MySQLInterfaceError as err:
    381     raise get_mysql_exception(
    382         msg=err.msg, errno=err.errno, sqlstate=err.sqlstate
    383     ) from err
```

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\mysql\connector\opentelemetry\context\_propagation.py:74, in **with\_context\_propagation.<locals>.wrapper**(cnx, \*args, \*\*kwargs)

```
    72 """Context propagation decorator."""
    73 if not OTEL_ENABLED or not cnx.otel_context_propagation:
--> 74     return method(cnx, *args, **kwargs)
    76 current_span = trace.get_current_span()
    77 tp_header = None
```

File ~\AppData\Local\Programs\Python\Python312\Lib\site-packages\mysql\connector\connection\_cext.py:669, in **CMySQLConnection.cmd\_query**(self, query, raw, buffered, raw\_as\_string)

```
    661     self._cmysql.query(
    662         query,
    663         raw=raw,
    (...)
```

```

666         query_attrs=self.query_attrs,
667     )
668 except MySQLInterfaceError as err:
--> 669     raise get_mysql_exception(
670         err.errno, msg=err.msg, sqlstate=err.sqlstate
671     ) from err
672 except AttributeError as err:
673     addr = (
674         self._unix_socket if self._unix_socket else f"{self._host}:{self._po
rt}"
675     )

```

**ProgrammingError: 1304 (42000): PROCEDURE AddBooking already exists**

In [11]: `cursor.callproc("AddBooking")`

Out[11]: `()`

In [12]: `cursor.callproc("ManageBooking")`

Out[12]: `()`

In [13]: `results=next(cursor.stored_results())`

In [14]: `dataset=results.fetchall()`

In [15]: `for column_id in cursor.stored_results():`  
`cols=[column[0] for column in column_id.description]`  
  
`for result in dataset:`  
`print("\t",cols[2],":",result[2], ' is already reserved ', 'on ', cols[3],":",`

```

TableNumber : 12 is already reserved on BookingDate : 2022-12-11
TableNumber : 12 is already reserved on BookingDate : 2022-12-10
TableNumber : 15 is already reserved on BookingDate : 2022-12-08
TableNumber : 19 is already reserved on BookingDate : 2022-12-09
TableNumber : 5 is already reserved on BookingDate : 2022-12-11
TableNumber : 6 is already reserved on BookingDate : 2022-12-10
TableNumber : 8 is already reserved on BookingDate : 2022-12-08
TableNumber : 9 is already reserved on BookingDate : 2022-12-15
TableNumber : 11 is already reserved on BookingDate : 2022-12-14
TableNumber : 17 is already reserved on BookingDate : 2022-12-05
TableNumber : 99 is already reserved on BookingDate : 2022-12-10

```

In [16]: `connection.commit()`

In [8]: `cursor.execute("DROP PROCEDURE IF EXISTS UpdateBooking;")`

In [9]: `# Create a stored procedure named UpdateBooking.`  
`cursor.execute("DROP PROCEDURE IF EXISTS UpdateBooking;")`  
`stored_procedure_query="""CREATE PROCEDURE UpdateBooking()`  
`BEGIN`  
 `UPDATE Bookings`  
 `SET BookingDate='2022-12-11'`  
 `WHERE TableNumber=99;`

```
END
"""
```

```
In [10]: cursor.execute(stored_procedure_query)
```

```
In [11]: cursor.callproc("UpdateBooking")
```

```
Out[11]: ()
```

```
In [12]: cursor.callproc("ManageBooking")
```

```
Out[12]: ()
```

```
In [13]: results=next(cursor.stored_results())
```

```
In [14]: dataset=results.fetchall()
```

```
In [15]: for column_id in cursor.stored_results():
          cols=[column[0] for column in column_id.description]

          for result in dataset:
              print("\t",cols[2],":",result[2], ' is already reserved ', 'on ', cols[3],":",

                    TableNumber : 12 is already reserved on BookingDate : 2022-12-11
                    TableNumber : 12 is already reserved on BookingDate : 2022-12-10
                    TableNumber : 15 is already reserved on BookingDate : 2022-12-08
                    TableNumber : 19 is already reserved on BookingDate : 2022-12-09
                    TableNumber : 5 is already reserved on BookingDate : 2022-12-11
                    TableNumber : 6 is already reserved on BookingDate : 2022-12-10
                    TableNumber : 8 is already reserved on BookingDate : 2022-12-08
                    TableNumber : 9 is already reserved on BookingDate : 2022-12-15
                    TableNumber : 11 is already reserved on BookingDate : 2022-12-14
                    TableNumber : 17 is already reserved on BookingDate : 2022-12-05
                    TableNumber : 99 is already reserved on BookingDate : 2022-12-11
```

```
In [16]: connection.commit()
```

```
In [19]: # Create a stored procedure named CancelBooking.
          cursor.execute("DROP PROCEDURE IF EXISTS CancelBooking;")
          stored_procedure_query="""CREATE PROCEDURE CancelBooking()
          BEGIN
              DELETE FROM Bookings

              WHERE TableNumber=99 and BookingDate='2022-12-11';

          END
          """
```

```
In [20]: cursor.execute(stored_procedure_query)
```

```
In [21]: cursor.callproc("CancelBooking")
```

Out[21]: ()

In [34]: `connection.commit()`

In [ ]: