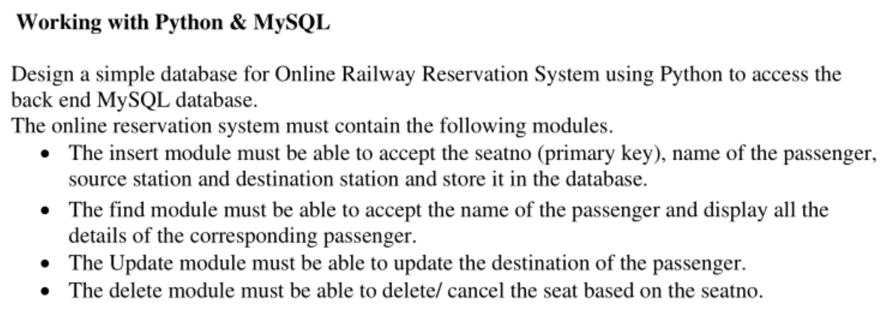
Lab 4 DBMS

106119029

Question 1



Working with Python & MySQL

Design a simple database for Online Railway Reservation System using Python to access the back end MySQL database.
The online reservation system must contain the following modules.

- The insert module must be able to accept the seatno (primary key), name of the passenger, source station and destination station and store it in the database.
- The find module must be able to accept the name of the passenger and display all the details of the corresponding passenger.
- The Update module must be able to update the destination of the passenger.
- The delete module must be able to delete/ cancel the seat based on the seatno.

Code

```python
from typing import Union
import psycopg2

class RailwayReservationSystem:
    def __init__(self, dbname:str, user:str, password:str,host:str,port:Union[str, int]):
        self.con = psycopg2.connect(
            host=host,
            database=dbname,
            user=user,
            password=password,
            port=int(port)
        )
        self.cur = self.con.cursor()
        self.create_table()

    def create_table(self):
        '''
    Schema for the table is:
    passengers {
            set_no: int, primary key,
            name: varchar(255),
            source: varchar(255),
            destination: varchar(255)
    }
    '''
    self.cur.execute('''DROP TABLE IF EXISTS passengers''')
    self.cur.execute('''CREATE TABLE IF NOT EXISTS
            passengers(seat_no INT PRIMARY KEY,
                name VARCHAR(255),
                source VARCHAR(255),
                destination VARCHAR(255))''')

# THis is the insert module
def insert(self, seat_no, name, source, destination):
    '''
    The insert module must be able to accept the seatno (primary key),
    name of the passenger, source station and destination
    station and store it in the database.
    '''
    self.cur.execute('''INSERT INTO
            passengers(seat_no, name, source, destination)
            VALUES(%s, %s, %s, %s)''', (seat_no, name, source, destination))

    print("INSERTED {} into table\n".format([seat_no,name,source,destination]))

# This is the find module
def find_and_display(self, name):
    '''
    The find module must be able to accept
    the name of the passenger and display all the
    details of the corresponding passenger.
    '''
    print("\nDetails for passenger with name: {}".format(name))
    self.cur.execute("SELECT * FROM passengers WHERE name = %s", (name,))
    self.print_cur_cursor()

def display_everything(self):
    print("Current Table:")
    self.cur.execute("SELECT * FROM passengers")
    self.print_cur_cursor()

# Update module
def update(self, seat_no, destination):
    '''
    The Update module must be able to
    update the destination of the passenger.
    '''
    print("BEFORE UPDATE\n");
    self.display_everything()
    self.cur.execute('''UPDATE passengers
            SET destination=%s
            WHERE seat_no = %s''', (destination, seat_no))
    print("\nUPDATED passenger with seat_no: {}\n".format(seat_no))
    print("AFTER UPDATE\n")
    self.display_everything()

# Delete module
def delete(self, seat_no):
    '''
    The delete module must be able
    to delete/ cancel the seat based on the seatno.
    '''
    self.cur.execute("DELETE FROM passengers WHERE seat_no = %s", (seat_no,))
    print("\nDELETED passenger with seat_no: {}\n".format(seat_no))


def print_cur_cursor(self):
    rows = self.cur.fetchall()
    # For formatting purposes
    print('\n{col0: ^15} | {col1: ^15} | {col2: ^15} | {col3: ^15}'
            .format(col0 ="seat_No",
                col1 = "name",
                col2 = "source",
                col3 = "destination"))
    print('{col0: <15} + {col1: <15} + {col2: <15} + {col3: <15}'
            .format(col0 = '-'*15,
                col1 = '-'*15,
                col2 = '-'*15,
                col3 = '-'*15))
    for row in rows:
        print('{col0: >15} | {col1: <15} | {col2: <15} | {col3: <15}'
                .format(col0 = row[0],
                    col1 = row[1],
                    col2 = row[2],
                    col3 = row[3]))

def close_conn(self):
    self.con.commit()
    self.con.close()
    print("Connection closed successfully")

def drop_table(self):
    self.cur.execute('drop table passengers');
    self.con.commit();

def print_barriers(self):
    print("="*80)


def main():
#  print(options)
reservation = RailwayReservationSystem('postgres',
        'postgres',
        '',
        'localhost',
        5432)

reservation.insert(1, 'Dipesh', 'Trichy', 'Kathmandu')
reservation.insert(2, 'Ram', 'Bangalore', 'Kolkata')
reservation.insert(3, 'Shyam', 'Delhi', 'Hyderabad')
reservation.insert(4, 'Sita', 'Chennai', 'Pune')
reservation.insert(5, 'Lakshmi', 'Pune', 'Kochi')
reservation.insert(6, 'Dipesh', 'Trichy', 'Kochi')

reservation.display_everything()

reservation.print_barriers()
reservation.find_and_display('Dipesh')
reservation.print_barriers()

reservation.update(2, 'Mumbai')
reservation.print_barriers()

reservation.delete(3)
reservation.print_barriers()

reservation.display_everything()
reservation.print_barriers()
#  reservation.drop_table()
reservation.close_conn()

main()
```

Output

```
INSERTED [1, 'Dipesh', 'Trichy', 'Kathmandu'] into table

INSERTED [2, 'Ram', 'Bangalore', 'Kolkata'] into table

INSERTED [3, 'Shyam', 'Delhi', 'Hyderabad'] into table

INSERTED [4, 'Sita', 'Chennai', 'Pune'] into table

INSERTED [5, 'Lakshmi', 'Pune', 'Kochi'] into table

INSERTED [6, 'Dipesh', 'Trichy', 'Kochi'] into table

Current Table:

    seat_No     |      name       |     source      |   destination
--------------- + --------------- + --------------- + ---------------
              1 | Dipesh          | Trichy          | Kathmandu
              2 | Ram             | Bangalore       | Kolkata
              3 | Shyam           | Delhi           | Hyderabad
              4 | Sita            | Chennai         | Pune
              5 | Lakshmi         | Pune            | Kochi
              6 | Dipesh          | Trichy          | Kochi
================================================================================

Details for passenger with name: Dipesh

    seat_No     |      name       |     source      |   destination
--------------- + --------------- + --------------- + ---------------
              1 | Dipesh          | Trichy          | Kathmandu
              6 | Dipesh          | Trichy          | Kochi
================================================================================
BEFORE UPDATE

Current Table:

    seat_No     |      name       |     source      |   destination
--------------- + --------------- + --------------- + ---------------
              1 | Dipesh          | Trichy          | Kathmandu
              2 | Ram             | Bangalore       | Kolkata
              3 | Shyam           | Delhi           | Hyderabad
              4 | Sita            | Chennai         | Pune
              5 | Lakshmi         | Pune            | Kochi
              6 | Dipesh          | Trichy          | Kochi
```

```
UPDATED passenger with seat_no: 2

AFTER UPDATE

Current Table:

    seat_No     |      name       |     source      |   destination
--------------- + --------------- + --------------- + ---------------
              1 | Dipesh          | Trichy          | Kathmandu
              3 | Shyam           | Delhi           | Hyderabad
              4 | Sita            | Chennai         | Pune
              5 | Lakshmi         | Pune            | Kochi
              6 | Dipesh          | Trichy          | Kochi
              2 | Ram             | Bangalore       | Mumbai
================================================================================

DELETED passenger with seat_no: 3

================================================================================
Current Table:

    seat_No     |      name       |     source      |   destination
--------------- + --------------- + --------------- + ---------------
              1 | Dipesh          | Trichy          | Kathmandu
              4 | Sita            | Chennai         | Pune
              5 | Lakshmi         | Pune            | Kochi
              6 | Dipesh          | Trichy          | Kochi
              2 | Ram             | Bangalore       | Mumbai
================================================================================
Connection closed successfully
```