

CMP-4010B Database Systems

Student Name:

Reg No:

Marker:

Before you begin running your SQL commands, delete all your own test data and insert my own test data (inserts2020.txt); use tests at the end of the document to ensure all data has been inserted correctly. Then proceed to run each task (1 to 8) as instructed below. Where your output is expected (in red), insert the SQL statement you run and then the output of running the statement on this document. It may be that a task requires you to run more than one SQL statement; in such cases insert in order the statements you run to achieve the task. You can insert any screenshots of running the tasks in PGAdmin as they will contain both the SQL and the output, but please make sure any text is of a size that can be read easily for marking. If the running of the SQL command results on an error reported by the SQL environment then record that error (i.e. copy it and paste the screenshot with the error shown). When asked, insert also the contents of particular tables (i.e. show the result of selecting all tuples from that table). If you have not implemented a particular task write 'NOT DONE' as the output. Please **do not delete** the marks allocated to each section (in blue) as those will be used by markers. An example of how to fill the form is in the appendix.

Testing task 1

1.b Given flight details create new flight record

- a) Create a new flight with values: flight ID = 108, origin = 'STN', destination = 'BHD', flight date = '29/7/2020', maximum capacity = 10, and price per seat = 70.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

- b) Create a new flight with values: flight ID = 109, origin = 'STN', destination = 'BHD', flight date = '29/7/2019', maximum capacity = 10, and price per seat = 70.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

- c) Create a new flight with values: flight ID = 110, origin = 'STN', destination = 'STN', flight date = '29/7/2020', maximum capacity = 10, and price per seat = 70.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

1.c Given passenger details create new passenger records

- a) Create a new passenger with values: PassengerID = 1025, FirstName = 'Ernesto', Surname = 'Picazo', PassportNo = '3434313', Nationality 'Spanish', DoB='30/7/1987'.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

- b) Create a new passenger with values: PassengerID = 1026, FirstName = 'Ernesto', Surname = 'Smith', PassportNo = '8383883', Nationality= 'British', DoB='30/2/2014'.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

DATE: Tuesday, 21 April 2020

Testing task 3

3. Check the availability of seats on all flights by showing the flight ID number, flight date along with the number of booked seats, number of available seats and maximum capacity.

a) Check availability on **ALL** flights.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /6

b) Check availability on a specific flight with **flightID = 103**.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

c) Check availability on specific flights with **destination = 'BRS'**.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

d) Check availability on specific flights with **date = '2020-07-24'**

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

Testing task 4

4. Check status of all seats for flightID = 101.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /4

Testing task 5

5. Produce Ranked list of lead customers by total spend.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /5

Testing task 6

6. Insert a flight booking.

- a) Check if a customer exists as part of inserting a new flight booking. Check for lead customer with customerID= 12 or with Surname 'Sayers'. Show results. If found, insert flight booking with details: BookingID = 513, customerID=12, FlightID=103 , NumSeats = 3 , Status = 'R'. If customer not found, then the process of inserting a flight booking should be cancelled.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /4

SHOW THE CONTENTS OF THE LEADCUSTOMER TABLE AND FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /2

- b) Insert a new lead customer as part of inserting a new flight booking. The details of the lead customer are customerID= 9, FirstName = 'Dan' , Surname = 'Sayers', Billing Address = '39a Morley Lane, Southampton', email = 'D.Smith@hotmail.com'. If the lead customer is inserted correctly, insert flight booking with details: BookingID = 514, customerID=9 , FlightID=103 , NumSeats = 3 , Status = 'R'. If there are any problems, then the process of inserting a flight booking should be cancelled.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /3

SHOW THE CONTENTS OF THE LEADCUSTOMER TABLE AND FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /2

- c) Insert a new lead customer as part of inserting a new flight booking. The details of the lead customer are customerID= 13, FirstName = 'Ben' , Surname = 'Morgan', Billing Address = '1 The Street, Norwich, email = 'b.morgan@hotmail.com'. If the lead customer is inserted correctly, insert flight booking with details: BookingID = 515, customerID=13, FlightID=103 , NumSeats = 3 , Status = 'R'. If there are any problems, then the process of inserting a flight booking should be cancelled.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /4

SHOW THE CONTENTS OF THE LEADCUSTOMER TABLE AND FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /2

- d) Insert a new lead customer as part of inserting a new flight booking. The details of the lead customer are customerID= 14, FirstName = 'Peter' , Surname = 'Brown', Billing Address = '3a Hill Street, Southampton', email = 'P.Brown@hotmail.com'. If the lead customer is inserted correctly, insert flight booking with details: BookingID = 516, customerID= 14, FlightID=103, NumSeats = 2 , Status = 'R'. If there are any problems, then the process of inserting a flight booking should be cancelled.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /4

SHOW THE CONTENTS OF THE LEADCUSTOMER TABLE AND FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /2

Testing task 7

7. Allocate seats to passengers

- a) Allocate seat for passenger 1024 in flight booking 510. The seat allocated is '8A'.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

SHOW THE CONTENTS OF THE SEATBOOKING TABLE AT THE END OF THE TASK

Marks: /1

- b) Allocate seat for passenger 1025 in flight booking 510. The seat allocated is '8B'.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

SHOW THE CONTENTS OF THE SEATBOOKING TABLE AT THE END OF THE TASK

Marks: /1

Testing task 8

8. Given bookingID cancel booking for BookingID= 509.

- a) Select from flight booking table to show it has been cancelled.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /2

SHOW THE CONTENTS OF THE FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /1

- b) Rerun task 4 (status of flights) for FlightID=104 as that has been affected.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

- c) Rerun task 5 (ranked list of lead customers) to see what has changed.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

Testing task 2

2. Delete a Lead Customer.

a) Delete lead customer with CustomerID=3.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

b) Run task 8 (cancel a booking) for bookingID 504.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

c) Delete lead customer with CustomerID=3.

INSERT YOUR SQL QUERY AND OUTPUT HERE

Marks: /1

SHOW THE CONTENTS OF THE LEADCUSTOMER TABLE AND FLIGHTBOOKING TABLE AT THE END OF THE TASK

Marks: /2

END OF TESTING

APPENDIX: EXAMPLE OF HOW TO FILL THIS FORM/DOCUMENT

Testing task 1

1. Given flight details create new flight record

- d) Create a new flight with values: flight ID = 120, origin = 'STN', destination = 'OVD', flight date = '30/7/2020', maximum capacity = 5, and price per seat = 100.

INSERT YOUR SQL QUERY AND OUTPUT HERE

The screenshot shows a SQL query editor interface. The top bar displays the user 'bli/bli@cmpstadb-01.cmp.uea.ac.uk'. Below the bar are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL code:

```
461  
462 --Create a new flight with values: flight ID = 120, origin = 'STN', destination = 'OVD',  
463 --flight date = '30/7/2020', maximum capacity = 5, and price per seat = 100.  
464 INSERT INTO Flight VALUES (120, '30/7/2020', 'STN', 'OVD', 5,100);  
465  
466  
467  
468
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Messages' tab is selected, showing the output: 'INSERT 0 1' and 'Query returned successfully in 170 msec.'

Marks: /1

SHOW THE CONTENTS OF FLIGHT TABLE AT THE END OF THE TASK

The screenshot shows a SQL query editor interface. The top bar displays the user 'bli/bli@cmpstadb-01.cmp.uea.ac.uk'. Below the bar are tabs for 'Query Editor' and 'Query History'. The query editor contains the following SQL code:

```
461  
462 --Create a new flight with values: flight ID = 120, origin = 'STN', destination = 'OVD',  
463 --flight date = '30/7/2020', maximum capacity = 5, and price per seat = 100.  
464 INSERT INTO Flight VALUES (120, '30/7/2020', 'STN', 'OVD', 5,100);  
465  
466 SELECT * FROM FLIGHT;  
467  
468
```

Below the query editor are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing the following table:

	flightid [PK] integer	flightdate timestamp without time zone	origin character varying (30)	destination character varying (30)	maxcapacity integer	priceperseat numeric
1	120	2 Editable column 0	STN	OVD	5	100

Marks: /1

*** Note: if you wish to submit evidence of running a Node.JS program, include it after the SQL output as an additional screenshot so show you achieved the task in Node.JS.***