

Supplementary Testing Document

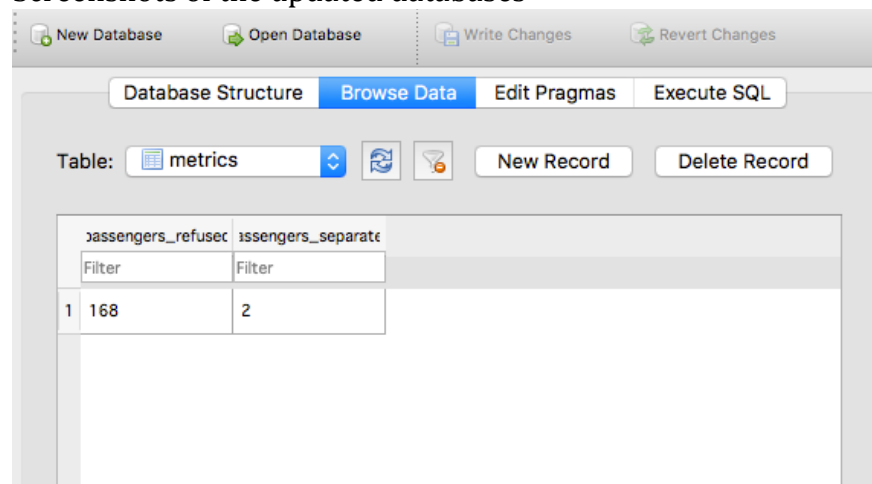
Testing the booking function.

The booking function allocates seats and updates the refused and separated metrics. To test this we created two versions of a database representing a full plane. One of these had the passengers refused and passengers separated metrics full and the other had these metrics empty.

Screenshot of running the program on the two databases

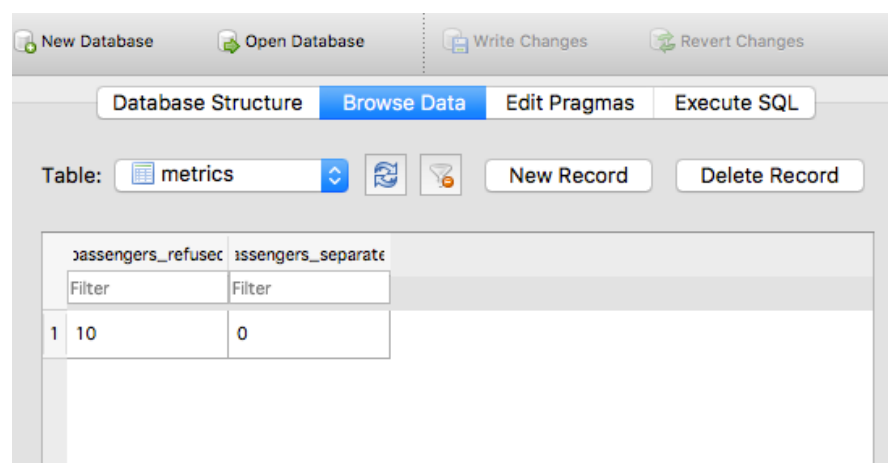
```
Marks-MacBook-Pro-3:Programming Assignment markdoyle$ python seat_assign_1245042]
8_13537123_16200240.py full_plane_metrics.db 10_passenger.csv
Bookings Processed
Passengers Refused: 168
Passengers Separated: 2
Marks-MacBook-Pro-3:Programming Assignment markdoyle$ python seat_assign_1245042]
8_13537123_16200240.py full_plane_no_metrics.db 10_passenger.csv
Bookings Processed
Passengers Refused: 10
Passengers Separated: 0
Marks-MacBook-Pro-3:Programming Assignment markdoyle$
```

Screenshots of the updated databases



The screenshot shows a database browser interface with a toolbar at the top containing 'New Database', 'Open Database', 'Write Changes', and 'Revert Changes'. Below the toolbar are tabs for 'Database Structure', 'Browse Data', 'Edit Pragmas', and 'Execute SQL'. The 'Browse Data' tab is selected, showing a table named 'metrics'. The table has two columns: 'passengers_refused' and 'passengers_separated'. The first row of data shows 168 refused passengers and 2 separated passengers.

	passengers_refused	passengers_separated
1	168	2



The screenshot shows the same database browser interface as above, but the 'metrics' table now shows 10 refused passengers and 0 separated passengers.

	passengers_refused	passengers_separated
1	10	0

The program is tested with on both a uniform plane layout where all rows have 4 seats and a layout where there are rows with 2 seats and others with 4.

Below is the resulting database with the program run on a plane with rows of both 2 and 4.

Database Structure Browse Data Edit Pragmas Execute SQL

Table: seating New Record Delete Record

	row ▲	seat	name
	Filter	Filter	Filter
1	1	A	Nicole McConville
2	1	C	Nicole McConville
3	2	A	Mark Doyle
4	2	C	
5	3	A	Madhura Kashikar
6	3	C	Madhura Kashikar
7	3	D	Madhura Kashikar
8	3	F	
9	4	A	Mark Doyle
10	4	C	Mark Doyle
11	4	D	Mark Doyle

|< < 1 - 11 of 24 > >|

Go to: 1

It can be seen from the screenshot above the program found the booking of 2 and assigned it to the row with two seats minimizing separation.

The rest of the functions are unit tested appropriately and can be found in a file entitled testing.py