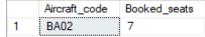# ADVANCED DATABASE – PRACTICE 1

Complete following tasks on Airline Reservation database.

1. Use CASE… WHEN… THEN… ELSE structure to write a query to display Passenger name, PP No, Aircraft_code, Journey_date, and WithCompanion from table Reservation and Passenger. The column WithCompanion should be 0 if the corresponding passenger books only one seat, otherwise it is 1.

	Name	PP No	Aircraft_code	Journey_date	WithCompanion
1	Allen Smith	117889	IC01	2001-03-15 00:00:00.000	1
2	Stella Smith	322901	IC01	2001-03-15 00:00:00.000	1
3	Pam Smith	279011	IC01	2001-03-15 00:00:00.000	1
4	Peter Jones	293211	BA02	2001-05-15 00:00:00.000	1
5	Lily Jones	347821	BA02	2001-05-15 00:00:00.000	1
6	James Crawford	123111	IC01	2001-03-15 00:00:00.000	1
7	Kitty Crawford	1237112	IC01	2001-03-15 00:00:00.000	1
8	Alex Lee	40103	BA02	2001-03-21 00:00:00.000	1
9	Greta Lee	41312	BA02	2001-03-21 00:00:00.000	1
10	Shania Lee	34784	BA02	2001-03-21 00:00:00.000	1
11	Christopher Lee	12453	BA02	2001-03-21 00:00:00.000	1
12	Stella Morris	31567	IC01	2001-03-15 00:00:00.000	1

2. From Reservation table, display Aircraft_code with its number of booked seats. The query only return rows having booked seats greater than 5. Hint: you should use HAVING clause
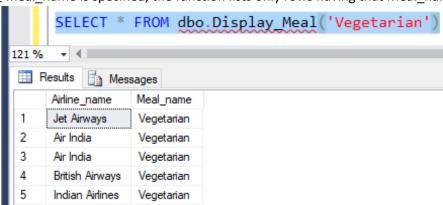
	Aircraft_code	Booked_seats
1	BA02	7

3. Display the record of table Reservation that having 2nd highest number_of_seats.

	PNR_no	Aircraft_code	Journey_date	Class_code	No_of_seats	Address	Contact_no	Status
1	2	BA02	2001-05-15 00:00:00.000	FC	3	Pushpa Vihar, 613 Defence colony, New Delhi	6453892	C

4. Create a view named vw_male_passenger containing PNR number (PRN_no), aircraft code, ticket number, passenger name, and reservation status for passengers with Sex = 'Male'

5. Show the data from the view vw_male_passenger.

6. Try updating the Sex of passenger with PNR_No = 4 to 'Gfluid'.

7. Modify the view vw_male_passenger so that it prevents modification of Sex. Hint: use WITH CHECK OPTION

8. Modify the view vw_male_passenger so that nobody can perform altering or deleting operations on its base tables that will make the view invalid.

9. Using sp_spaceused, check the storage space occupied by table Passenger and view vw_male_passenger

10. Create a stored procedure named process_counter to display the square root value of integer numbers from 10 to 1 in descending order.

11. Basing on task #1, create a stored procedure named display_passenger_companion that accepts Aircraft_code as a parameter. Given a value of @Aircraft_code, the procedure will show the corresponding Passenger name, PP No, Aircraft_code, Journey_date, and WithCompanion

12. Write a procedure named Insert_Passenger with appropriate parameters to insert a record into Passenger table. If the value of age is negative, an error should be raised, and the procedure does not insert data.

13. Create a function named Count_non_chinese_meal () without parameter to display number of meals without 'Chinese' in the name.

14. Call Count_non_chinese_meal () function.

15. Create a function named Display_meal that accepts @meal_name as a parameter and display which Meal_name is served on which Airline_name. It must work as follows:
    a. If the @meal_name is specified, the function lists only rows having that Meal_name

```
SELECT * FROM dbo.Display_Meal('Vegetarian')
```

121 %

Results | Messages

	Airline_name	Meal_name
1	Jet Airways	Vegetarian
2	Air India	Vegetarian
3	Air India	Vegetarian
4	British Airways	Vegetarian
5	Indian Airlines	Vegetarian

    b. If '*' is passed as the value of @meal_name, the function returns all Meal_names associated with their Airline_names.

	Airline_name	Meal_name
1	Jet Airways	Vegetarian
2	Jet Airways	Non-Vegetarian
3	Air India	Vegetarian
4	Air India	Non-Vegetarian
5	Air India	Vegetarian
6	Air India	Chinese Vegetarian
7	Air India	Chinese Non-Vegetarian
8	Air India	Continental Vegetarian
9	Air India	Continental Non-Vegetarian
10	British Airways	Vegetarian
11	British Airways	Non-Vegetarian

16. Call Display_meal with @meal_name = 'Vegetarian' first, then try @meal_name = '*'.

17. Create a DELETE trigger that prevents deleting more than 1 rows from table Passenger at a time.

18. Write an INSERT trigger named ins_reservation on Reservation table. It goal is to ensure the referential integrity, that is, the inserted values of PNR_no, Aircraft_code, and Class_code must exist in tables Passenger, Flight, and Class_master respectively.

19. Write another INSERT trigger named ins_reservation_data on Reservation table. It does not allow insert Journey_date values smaller than current time. Hint: built-in function GETDATE() returns current datetime of the system.
20. Create an UPDATE trigger that does not allow someone change No_of_sets to a value greater than 5.