DBMS Project Report

PES University

Database Management Systems

UE18CS252

Submitted By

PES2201800013 Akshaya J

The AirlineReservation database is a management system that uses database technology to construct, store and manipulate various kinds of data related to booking or reserving of flight seats.

This database contains 12 tables: Airport, Client, Booking, Airlines, Flight status, Flight, Aircraft, Components, Manufacturer, Provides, Travel Class, Aircraft Seat.

The trigger implemented in this database checks if the departure time is earlier than the arrival time. If it is, the trigger throws an error and does not let the user update/add that data to the table.

The queries implemented in this project are: correlated nested query (1), aggregate query (3) and outer-join query (1).

This project aims at computerizing the manual process of a storing the details of passengers after they've booked a flight. It helps keep track of the passengers and the flights travelling from one place to another.

The data stored on the database can be used for further data analysis that will help in studying the travel plans and movement of people around the world.

Introduction:

Airline reservation system is an application of database management system which is used for booking and scheduling flight details. It is an integral part of today's world. It is especially helpful for customers looking into online booking.

It manages all the information about the customer, booking enquiry, reservation and so on. Maintaining and keeping track of various such information through the physical hard copies will prove to be very difficult. This will also pose the problem of inadequate security when it comes to protecting the data. In such a situation, the airline reservation system database can be used to store all this information securely and at one stretch.

The entities used are:

1. **Airport:** It maintains a record of the airport id, name and the city and the country in which the airport is located.

2. **Client:** It maintains a record of the client id, their first and last names and passport number.

3. **Booking:** Maintains a record of the booking number, price and the id of the client who's booking the ticket(s).

4. **Airlines:** Maintains a record of airlines id and main office.

5. **Flight status:** Maintains a record of status id, departure time, departure date, arrival time and arrival date.

6. **Flight:** Maintains a record of flight id, origin airport id, destination airport id, status id, airlines id and return flight id.

7. **Aircraft:** Maintains a record of aircraft id, model of the aircraft and the flight id.

8. **Components:** Maintains a record of components id and the name of the components.

9. **Manufacturer:** Maintains a record of manufacturer id and the name of the manufacturer.

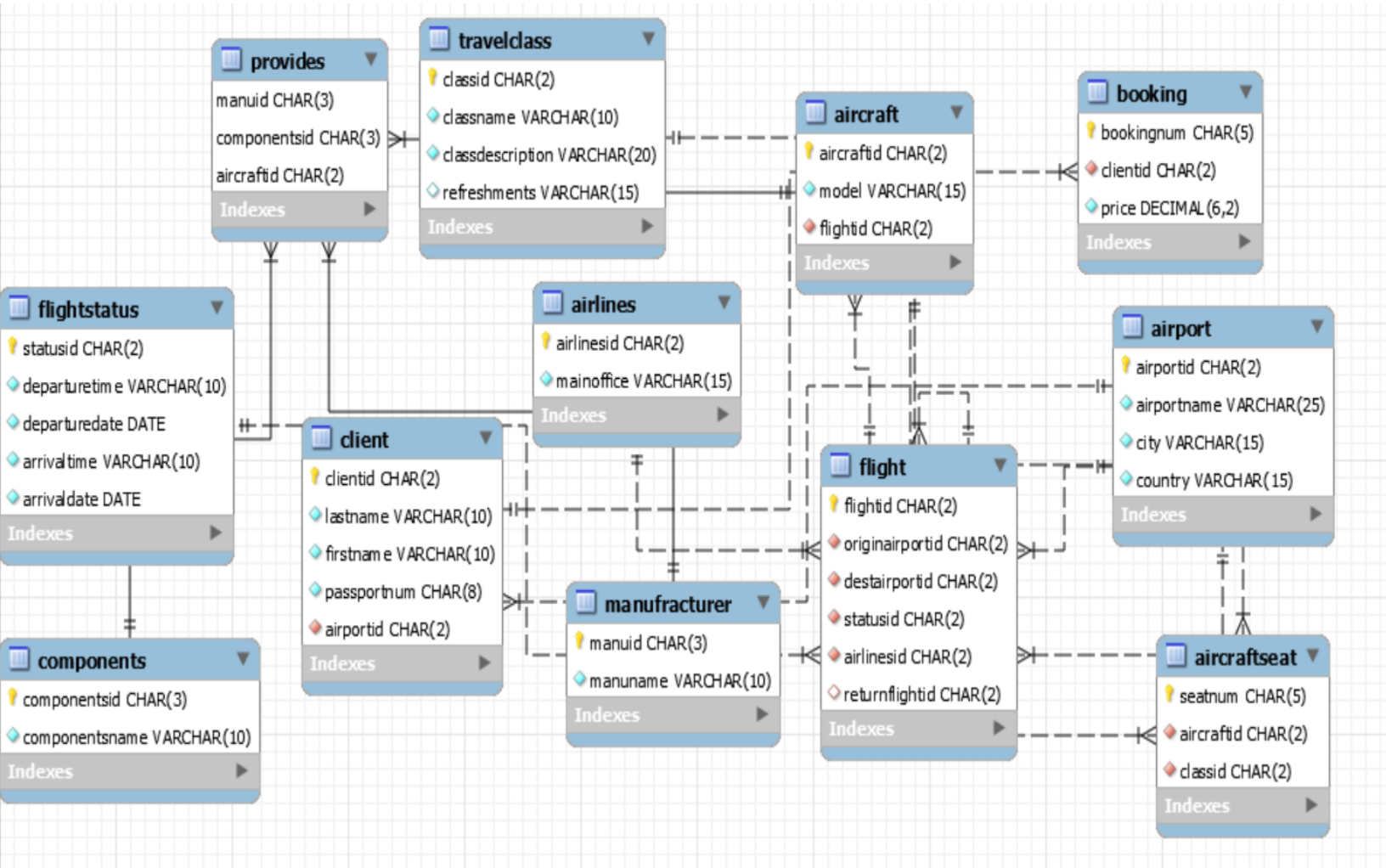
10. **Provides:** Maintains a record of the ids of the manufacturer, components and aircraft.

11. **Travel Class:** Maintains a record of class id, class name, class description and refreshments.

12. **Aircraft Seat:** Maintains a record of seat number, aircraft id, class id.

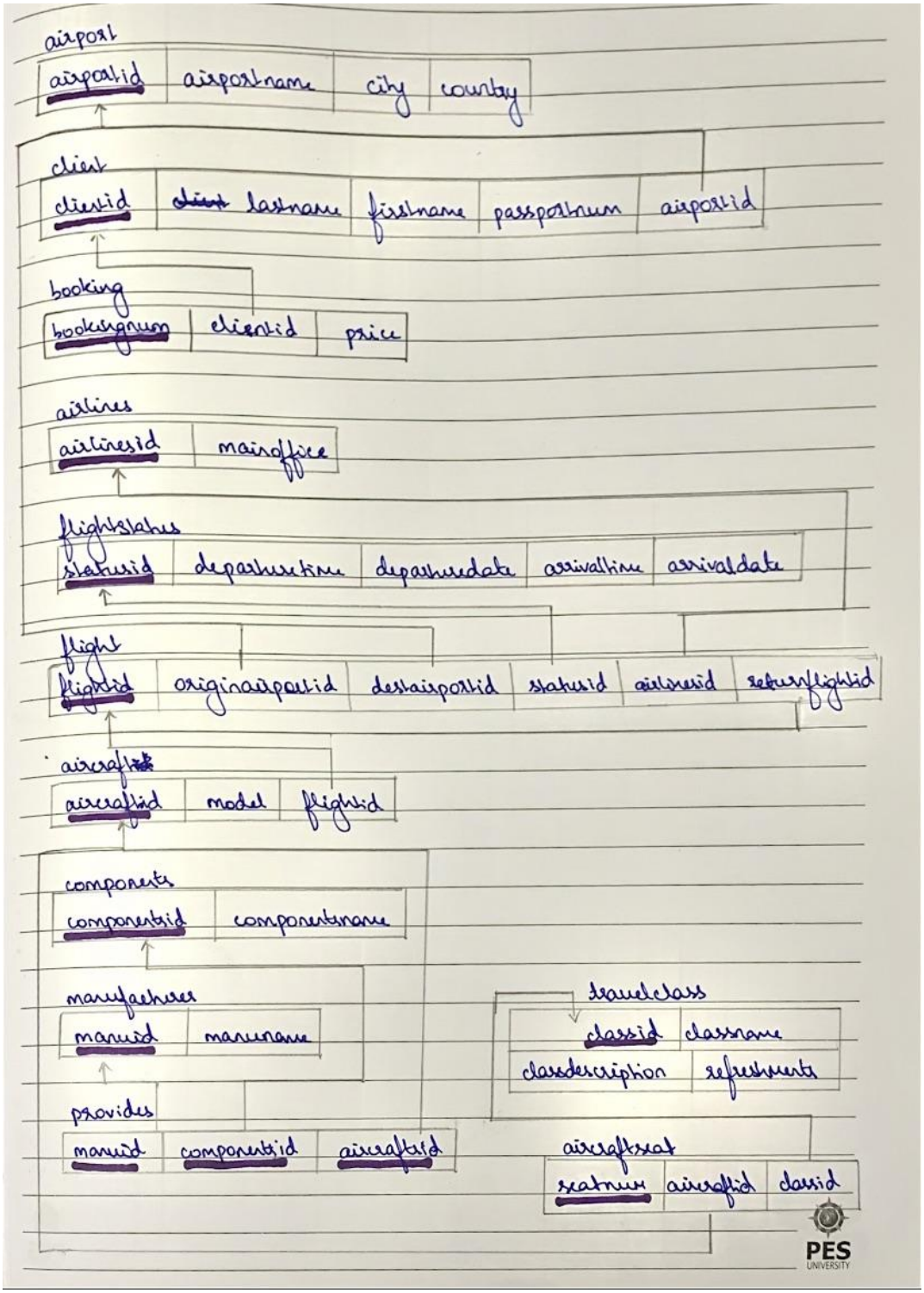
Data Model

ER MODEL:

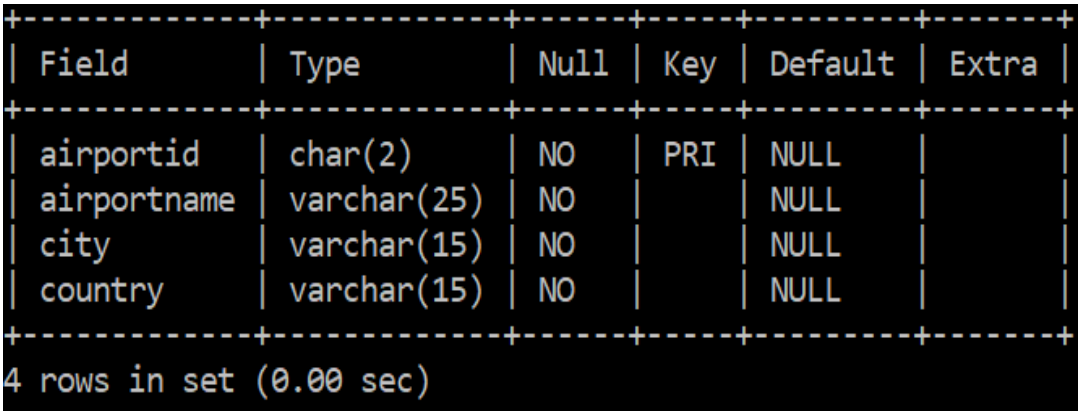


SCHEMA:

Relational Schema:

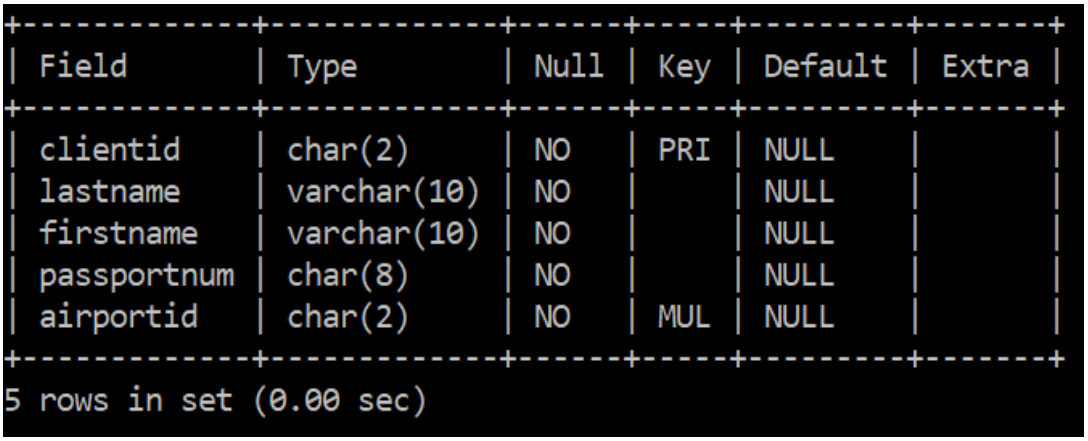


1. Table airport:



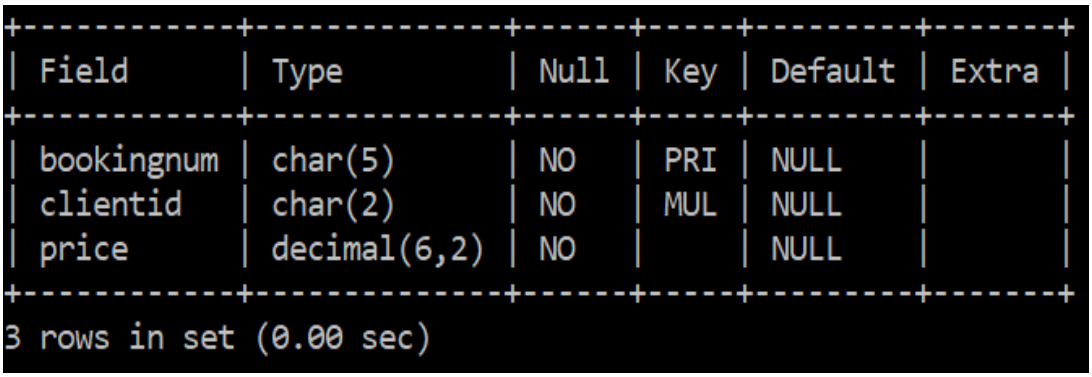
```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| airportid   | char(2)     | NO   | PRI | NULL    |       |
| airportname | varchar(25) | NO   |     | NULL    |       |
| city        | varchar(15) | NO   |     | NULL    |       |
| country     | varchar(15) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

2. Table client:



```
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| clientid    | char(2)     | NO   | PRI | NULL    |       |
| lastname    | varchar(10) | NO   |     | NULL    |       |
| firstname   | varchar(10) | NO   |     | NULL    |       |
| passportnum | char(8)     | NO   |     | NULL    |       |
| airportid   | char(2)     | NO   | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

3. Table booking:



```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| bookingnum | char(5)     | NO   | PRI | NULL    |       |
| clientid   | char(2)     | NO   | MUL | NULL    |       |
| price      | decimal(6,2) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

4. Table airlines:

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| airlinesid | char(2)     | NO   | PRI | NULL    |       |
| mainoffice | varchar(15) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

5. Table flightstatus:

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| statusid      | char(2)     | NO   | PRI | NULL    |       |
| departuretime | varchar(10) | NO   |     | NULL    |       |
| departuredate | date        | NO   |     | NULL    |       |
| arrivaltime   | varchar(10) | NO   |     | NULL    |       |
| arrivaldate   | date        | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

6. Table flight:

```
+-----------------+---------+------+-----+---------+-------+
| Field           | Type    | Null | Key | Default | Extra |
+-----------------+---------+------+-----+---------+-------+
| flightid        | char(2) | NO   | PRI | NULL    |       |
| originairportid | char(2) | NO   | MUL | NULL    |       |
| destairportid   | char(2) | NO   | MUL | NULL    |       |
| statusid        | char(2) | NO   | MUL | NULL    |       |
| airlinesid      | char(2) | NO   | MUL | NULL    |       |
| returnflightid  | char(2) | YES  | MUL | NULL    |       |
+-----------------+---------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

7. Table aircraft:

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| aircraftid | char(2)     | NO   | PRI | NULL    |       |
| model      | varchar(15) | NO   |     | NULL    |       |
| flightid   | char(2)     | NO   | MUL | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

8. Table components:

```
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| componentsid  | char(3)     | NO   | PRI | NULL    |       |
| componentsname | varchar(10) | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

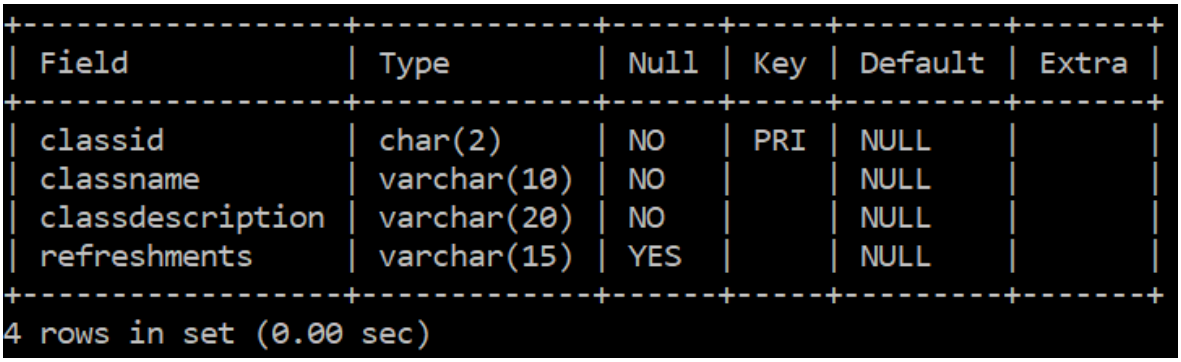
9. Table manufacturer:

```
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| manuid   | char(3)     | NO   | PRI | NULL    |       |
| manuname | varchar(10) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

10. Table provides:

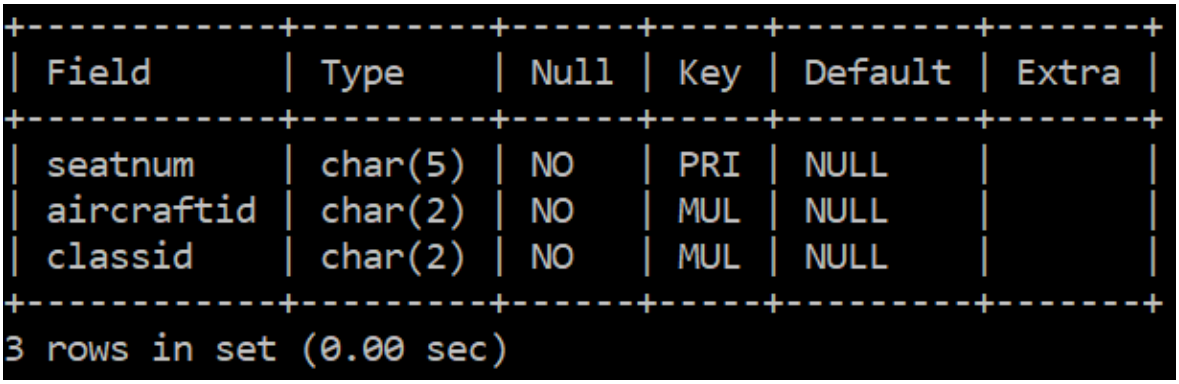
```
+--------------+---------+------+-----+---------+-------+
| Field        | Type    | Null | Key | Default | Extra |
+--------------+---------+------+-----+---------+-------+
| manuid       | char(3) | NO   | PRI | NULL    |       |
| componentsid | char(3) | NO   | PRI | NULL    |       |
| aircraftid   | char(2) | NO   | PRI | NULL    |       |
+--------------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

11. Table travelclass:



```
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| classid          | char(2)     | NO   | PRI | NULL    |       |
| classname        | varchar(10) | NO   |     | NULL    |       |
| classdescription | varchar(20) | NO   |     | NULL    |       |
| refreshments     | varchar(15) | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

12. Table aircraftseat:



```
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| seatnum    | char(5) | NO   | PRI | NULL    |       |
| aircraftid | char(2) | NO   | MUL | NULL    |       |
| classid    | char(2) | NO   | MUL | NULL    |       |
+------------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

FD and Normalization

FD:

Relation airport (airportid, airportname, city, country)
airportid -> airportname, city, county
city -> country

Relation client (clientid, lastname, firstname, passportnum, airportid)
clientid -> lastname, firstname, passportnum, airportid

Relation booking (bookingnum, clientid, price)
bookingnum -> clientid, price

Relation airlines (airlinesid, mainoffice)
airlinesid -> mainoffice

Relation flightstatus (statusid, departuretime, departuredate, arrivaltime, arrivaldate)
statusid -> departuretime, departuredate, arrivaltime, arrivaldate
departuretime -> departuredate, arrivaltime, arrivaldate

Relation flight (flightid, originairportid, destairportid, statusid, airlinesid, returnflightid)
flightid -> originairportid, destairportid, statusid, airlinesid, returnflightid

Relation aircraft (aircraftid, model, flightid)
aircraftid -> model, flightid

Relation components (componentsid, componentsname)
componentsid -> componentsname

Relation manufacturer (manuid, manuname)
manuid -> manuname

Relation travelclass (classid, classname, classdescription, refreshments)
classid -> classname, classdescription, refreshments
classname -> classdescription

Relation aircraftseat (seatnum, aircraftid, classid)
seatnum -> aircraftid, classid

NORMALISATION:

1st Normal Form:

Rules of 1st NF:

1. Each table cell should contain a single value.
2. Each record needs to be unique.

In my database, each entry in all the tables has a singular value. All the attributes are single valued. Therefore, no table breaks 1st Normal form.

2nd Normal Form:

Rules of 2nd NF:

1. Be in 1NF.
2. Single Column Primary Key.

In my database:

- If the *components* table and the *provides* table are merged, this violates the 2NF form since some of the attributes in the merged table are dependent on one of the primary keys and not both.

- If the *manufacturer* table and the *provides* table are merged, this violates the 2NF form since some of the attributes in the merged table are dependent on one of the primary keys and not both.

3rd Normal Form:

Rules of 3rd NF:

1. Be in 2NF.
2. Has no transitive functional dependencies.

In my database, if I merge the tables *aircraftseat* and *travelclass*, the foreign key, class id, will cease to exist and the derived table will violate 3rd NF form.

This happens because there's a transitive functional dependency between class name and class description (changing one changes the other and neither are primary keys).

DDL

```
create database AirlinesReservation;

use AirlinesReservation;

create TABLE airport
(          airportid             char(2)              NOT NULL,
           airportname           varchar(25)          NOT NULL,
           city                  varchar(15)          NOT NULL,
           country               varchar(15)          NOT NULL,
           PRIMARY KEY (airportid) );

select * from airport;

create TABLE client
(          clientid              char(2)              NOT NULL,
           lastname              varchar(10)          NOT NULL,
           firstname             varchar(10)          NOT NULL,
           passportnum           char(8)              NOT NULL,
           airportid             char(2)              NOT NULL,
           PRIMARY KEY (clientid),
           FOREIGN KEY (airportid) REFERENCES airport(airportid) );

select * from client;

create TABLE booking
(          bookingnum            char(5)              NOT NULL,
           clientid              char(2)              NOT NULL,
           price                 numeric(6,2)         NOT NULL,
           PRIMARY KEY (bookingnum),
           FOREIGN KEY (clientid) REFERENCES client(clientid) );

select * from booking;

create TABLE airlines
(          airlinesid      char(2)              NOT NULL,
           mainoffice      varchar(15)          NOT NULL,
           PRIMARY KEY (airlinesid)  );

select * from airlines;


create TABLE flightstatus
```

```
(          statusid              char(2)              NOT NULL,
           departuretime         varchar(10)          NOT NULL,
           departuredate         DATE            NOT NULL,
           arrivaltime           varchar(10)          NOT NULL,
           arrivaldate           DATE            NOT NULL,
           PRIMARY KEY (statusid)  );

select * from flightstatus;

create TABLE flight
(          flightid              char(2)              NOT NULL,
           originairportid       char(2)              NOT NULL,
           destairportid         char(2)              NOT NULL,
           statusid              char(2)              NOT NULL,
           airlinesid            char(2)              NOT NULL,
           returnflightid        char(2),
           PRIMARY KEY(flightid),
           FOREIGN KEY (originairportid) REFERENCES
airport(airportid),
           FOREIGN KEY (destairportid) REFERENCES airport(airportid),
           FOREIGN KEY (statusid) REFERENCES flightstatus(statusid),
           FOREIGN KEY (airlinesid) REFERENCES airlines(airlinesid),
           FOREIGN KEY (returnflightid) REFERENCES flight(flightid) );

select * from flight;

create TABLE aircraft
(          aircraftid      char(2)              NOT NULL,
           model           varchar(15)          NOT NULL,
           flightid        char(2)              NOT NULL,
           PRIMARY KEY (aircraftid),
           FOREIGN KEY (flightid) REFERENCES flight(flightid) );

select * from aircraft;

create TABLE components
(          componentsid          char(3)              NOT NULL,
           componentsname        varchar(10)          NOT NULL,
           PRIMARY KEY (componentsid) );

select * from components;

create TABLE manufacturer
(          manuid                char(3)              NOT NULL,
           manuname              varchar(10)          NOT NULL,
```

```
          PRIMARY KEY (manuid) );

select * from manufacturer;

create TABLE provides
(         manuid               char(3)              NOT NULL,
          componentsid         char(3)              NOT NULL,
          aircraftid           char(2)              NOT NULL,
          PRIMARY KEY (manuid, componentsid, aircraftid),
          FOREIGN KEY (manuid) REFERENCES manufacturer(manuid),
          FOREIGN KEY (componentsid) REFERENCES
components(componentsid),
          FOREIGN KEY (aircraftid) REFERENCES aircraft(aircraftid) );

select * from provides;

create TABLE travelclass
(         classid              char(2)              NOT NULL,
          classname            varchar(10)          NOT NULL,
          classdescription     varchar(20)          NOT NULL,
          refreshments         varchar(15),
          PRIMARY KEY (classid)   );

select * from travelclass;

create TABLE aircraftseat
(         seatnum              char(5)              NOT NULL,
          aircraftid           char(2)              NOT NULL,
          classid              char(2)              NOT NULL,
          PRIMARY KEY (seatnum),
          FOREIGN KEY (classid) REFERENCES travelclass(classid),
          FOREIGN KEY (aircraftid) REFERENCES aircraft(aircraftid)
);

select * from aircraftseat;
```

TRIGGERS:

```
CREATE DEFINER = CURRENT_USER TRIGGER
`airlinesreservation`.`flightstatus_BEFORE_INSERT` BEFORE INSERT ON
`flightstatus` FOR EACH ROW
BEGIN
Declare msg varchar(200);
     IF new.departuredate < new.arrivaldate then
          set msg = concat('flightstatus_BEFORE_INSERT: Departure
date earlier than arrival date', cast(new.statusid as char));
          signal sqlstate '45000' set message_text = msg;
End if;
END
```

This trigger checks if the departure time is earlier than the arrival time. If it is, the trigger throws an error and does not let the user update/add that data to the table.

```
1 •  use airlinesreservation;
2 •  INSERT INTO flightstatus VALUES('A1','17:30', '2019-01-01', '5:30', '2019-02-01');
```

Output:

SQL QUERIES:

CORRELATED NESTED QUERY:

Query One:

```
SELECT clientid, lastName, firstName from client WHERE airportid =
(SELECT airportid from client where firstname='Brantly' and lastName=
'Morgan');
```

In this query, we get a table with clientid, lastName and firstName as columns and the table contains the rows (client details) which have the same airport id as the client Brantly Morgan.

clientid	lastName	firstName
07	MORGAN	BRANTLY
10	Mary	Ellen
NULL	NULL	NULL

AGGREGATE QUERY: (using count)

Query One:

```
SELECT flightid from flight where destairportid IN (SELECT
destairportid FROM flight GROUP BY destairportid HAVING
COUNT(destairportid) > 4); /* Also considered as a correlated nested query */
```

In this query, we group the four or more flights which have the same destination airport id.

flightid
FF
FH
FI
FO
NULL

Query Two:

```
SELECT MAX (price) FROM booking;
```

	MAX(price)
▶	4125.27

In this query, the maximum booking price is selected.

Query Three:

```
SELECT MIN (price) FROM booking;
```

	MIN(price)
▶	1502.33

In this query, the maximum booking price is selected.

OUTER JOIN QUERY:

Query One:

```
      SELECT clientid, lastname, firstname, passportnum, C.airportid,
airportname, city, country
      FROM client C
      LEFT OUTER JOIN airport A
      ON C.airportid = A.airportid;
```

This query joins the tables client and airport using outer join.

	clientid	lastname	firstname	passportnum	airportid	airportname	city	country
▶	01	MARK	CUBAN	17642181	A1	LAX	Los Angeles	USA
	02	JIM	BEAM	18042151	A2	IAH	Atlanta	USA
	03	GEORGE	WASHINGTON	31532182	A3	IAH	Houston	USA
	04	PAUL	HEYMAN	43624111	A4	DFW	Dallas	USA
	05	LUCI	JOHNSON	92222085	A5	JFK	NewYork	USA
	06	MARIA	FRIGGIE	92015583	A6	DEN	Denver	USA
	07	MORGAN	BRANTLY	03532180	A7	MIA	Miami	
	08	LUKAS	PODOLSKI	08902180	A8	ORD	Chicago	USA
	09	AB	DEVILLIERS	03534321	A9	MSY	New Orleans	USA
	10	Mary	Ellen	12345678	A7	MIA	Miami	USA

CONCLUSION:

This project aims at computerizing the manual process of a tracking movement of flights and passengers.

Capabilities:

1. This database keeps track of the number of passengers/clients.
2. It keeps track of various flights and their travel log.
3. It keeps track of the airports and their schedules.
4. It keeps a record of the dates and times each flight operated on.
5. It also keeps track of the flight parts and its manufacturer.

Limitations:

1. This database contains no record of discounts, promos or advertising schemes. It does not keep a track of the clients' phone numbers.

Future enhancements:

1. The data stored on the database can be used for further data analysis that will help in studying the travel plans and movement of people around the world and their travel behaviour.
2. The data stored in this database can help the tourism industry work more efficiently.