



湖北工業大學
HUBEI UNIVERSITY OF TECHNOLOGY

DBMS

Course Design Report

Design Topic Online Airplane Reservation System

Mentor Prof. Jianxia Chen

Profession Computer Science & Technology

Class 19lq

Student ID 1911521213

Name MUNTASIR MD FAHIM

Date 2021-12-19

Catalog

- 1. Objective and significance of the project**
- 2. System operation environment description**
- 3. System requirements analysis**
 - 3.1 Organization chart**
 - 3.2 Data flow graph and data dictionary**
 - 3.3 System function structure**
- 4. Database design**
 - 4.1 Conceptual design (E-R diagram)**
 - 4.2 Logical structure design**
 - 4.3 Physical structure design**
- 5. Database implementation and operation**
 - 5.1 Introduction of database management system**
 - 5.2 Database creation and data entry**
 - 5.3 Database query**
 - 5.4 Database control**
- 6. Program design description**
 - 6.1 Database connection**
 - 6.2 System menu**
 - 6.3 User login**
 - 6.4 Data entry, modification and deletion**
 - 6.5 Database query**
- 7. Summary of course design**
 - 7.1 Problems and solutions in curriculum design**
 - 7.2 Analysis of existing problems**
 - 7.3 Experience of curriculum design**

1. Objective and significance of the project

The web based “airline reservation system” project is an attempt to stimulate the basic concepts of airline reservation system. The system enables the customer to do the things such as search for airline flights for two travel cities on a specified date, choose a flight based on the details, reservation of flight and cancellation of reservation.

The system allows the airline passenger to search for flights that are available between the two travel cities, namely the “Departure city” and “Arrival city” for a particular departure and arrival dates. The system displays all the flight’s details such as flight no, name, price and duration of journey etc.

After search the system display list of available flights and allows customer to choose a particular flight. Then the system checks for the availability of seats on the flight. If the seats are available then the system allows the passenger to book a seat. Otherwise it asks the user to choose another flight.

To book a flight the system asks the customer to enter his details such as name, address, city, state, credit card number and contact number. Then it checks the validity of card and book the flight and update the airline database and user database. The system also allows the customer to cancel his/her reservation, if any problem occurs.

2. System Operation Environment Description

Develop an Internet airline reservation system in two different ways.

Both systems shall be client server systems. The first is to apply all the business logic in the thin client and server. Then, the system should be refactored into a system with thick clients, in which as much business logic as possible is executed by the clients. When a customer opens the client page, there should be a way to select the city of departure and destination and the date of travel. You can also choose an airline. The system should be able to find and Book itineraries by merging flights from different airlines. Each airline has its own database. When merging flights, the time required for connecting airports should be considered, i.e. connection time. When booking all flights of the same airline, the price will be discounted according to the airline's own discount (%).

3. System Requirements Analysis

The system shall have the following use cases:

1. find the best (price or time) itinerary between a and B, including the following use cases:

- *Choose the city of departure,*
- *Select the destination city,*
- *Select the date,*
- *Choose to sort by price or by travel time,*
- *Choose the number of people. Only consider flights with enough seats.*

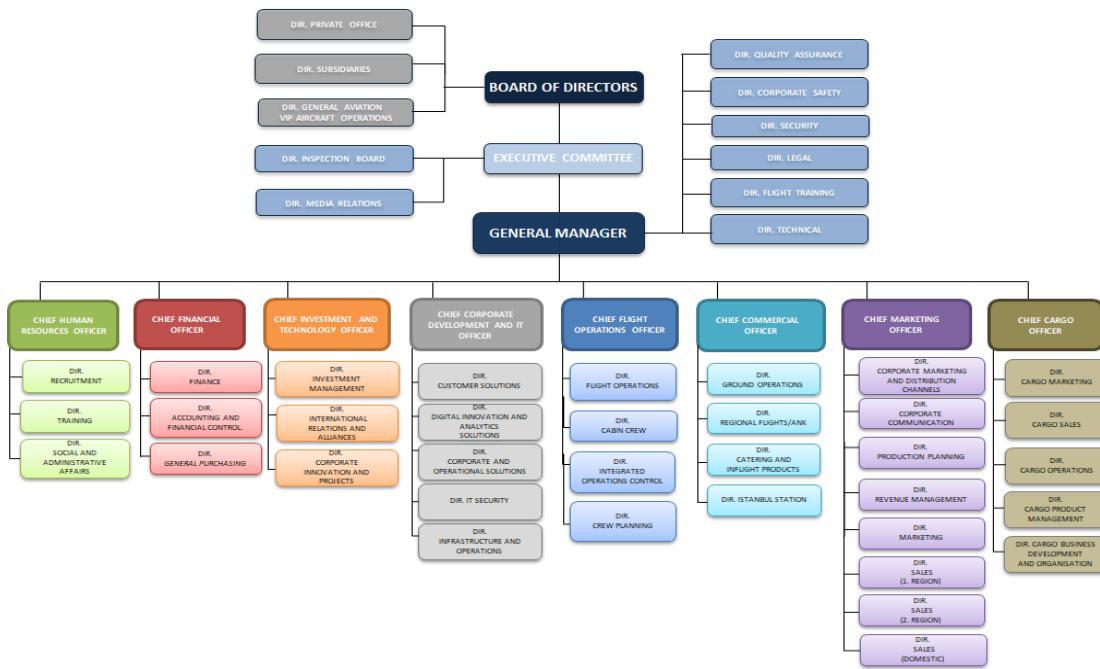
2. Use the following sub use case to book travel:

- *select trip from the previous use case,*
- *select seat preference (window, aisle or middle),*
- *input passenger data,*
- *submit the booking and get a receipt.*

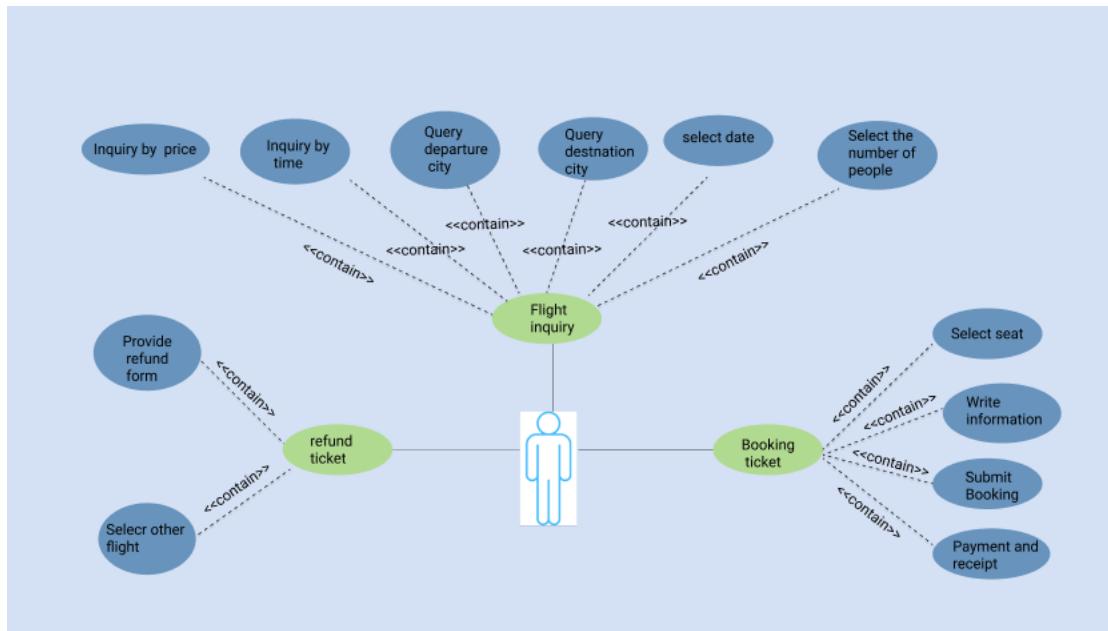
3. Cancel the trip.

Other use cases can also be considered

3.1 Organization chart



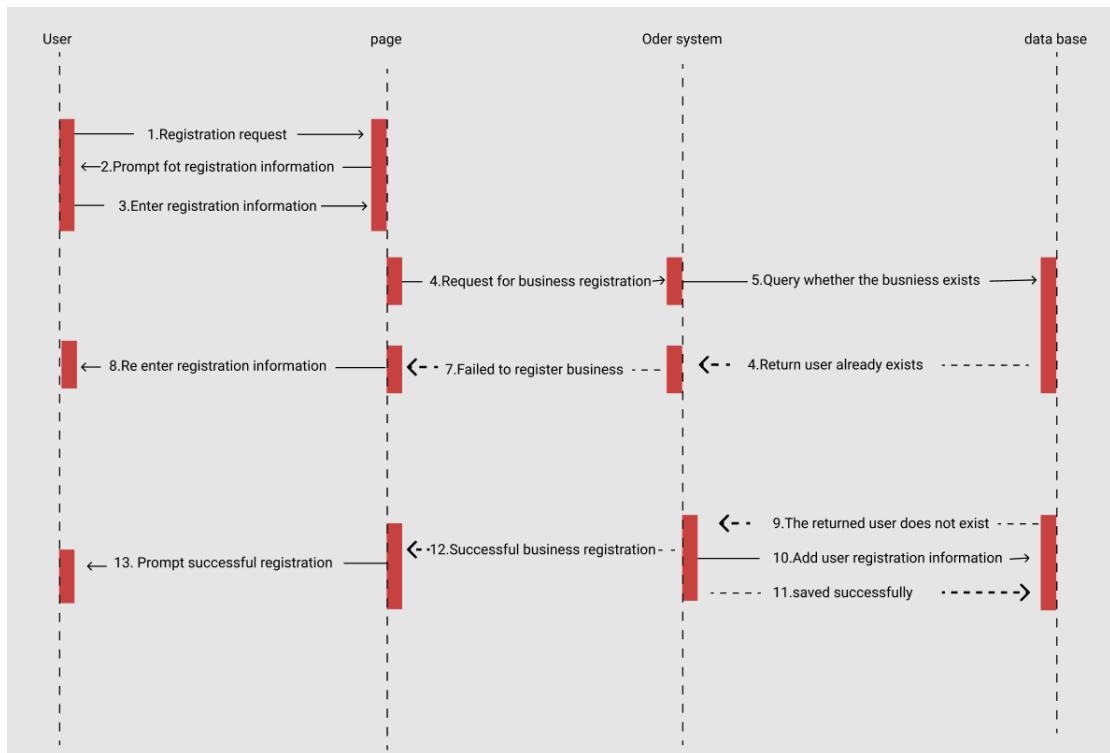
3.2 Data flow graph and data dictionary



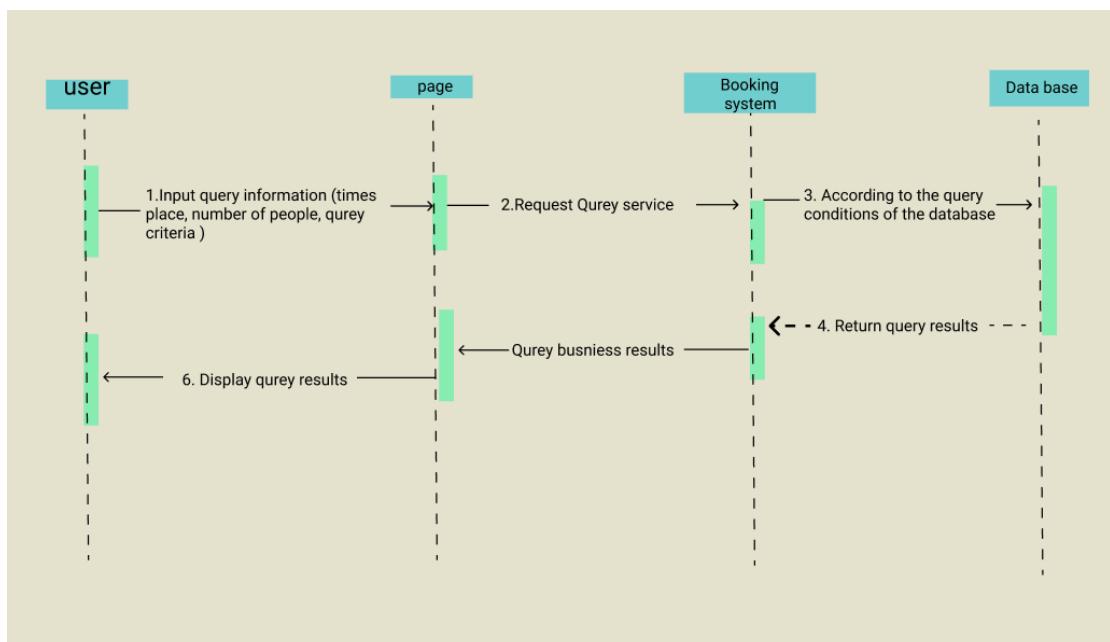
3.3 System function structure

3.3.1 Three layer C / S structure

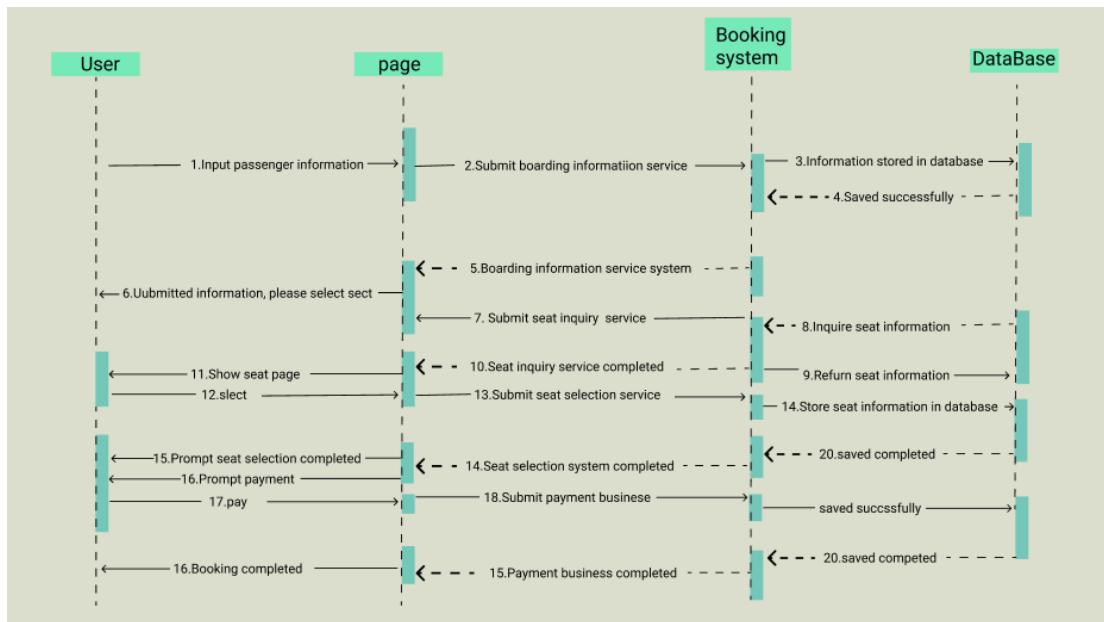
User registration sequence diagram:



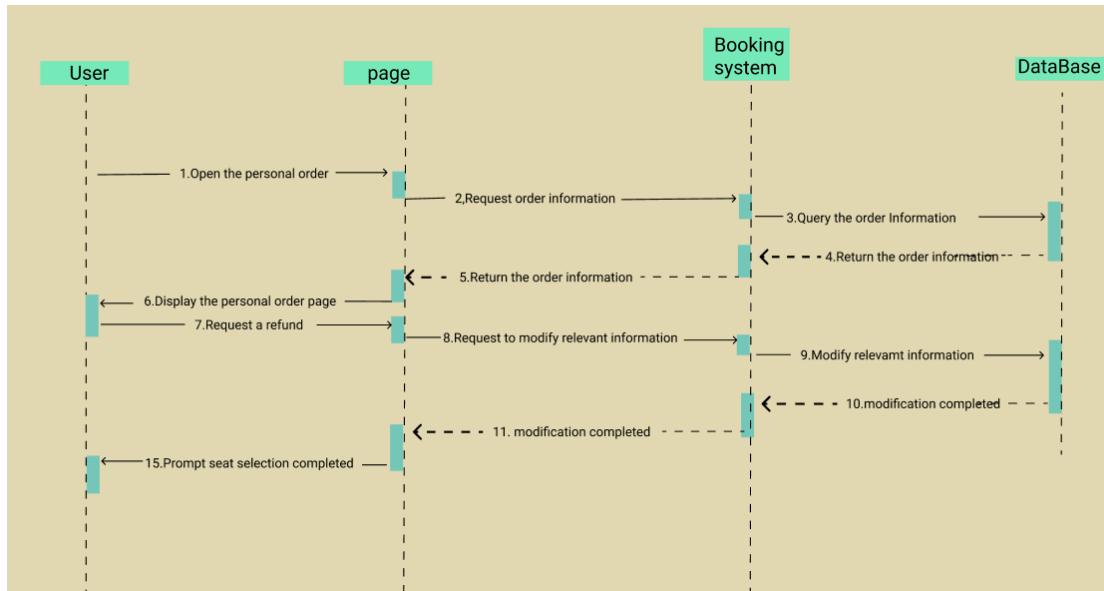
Flight query sequence diagram:



User booking sequence diagram:

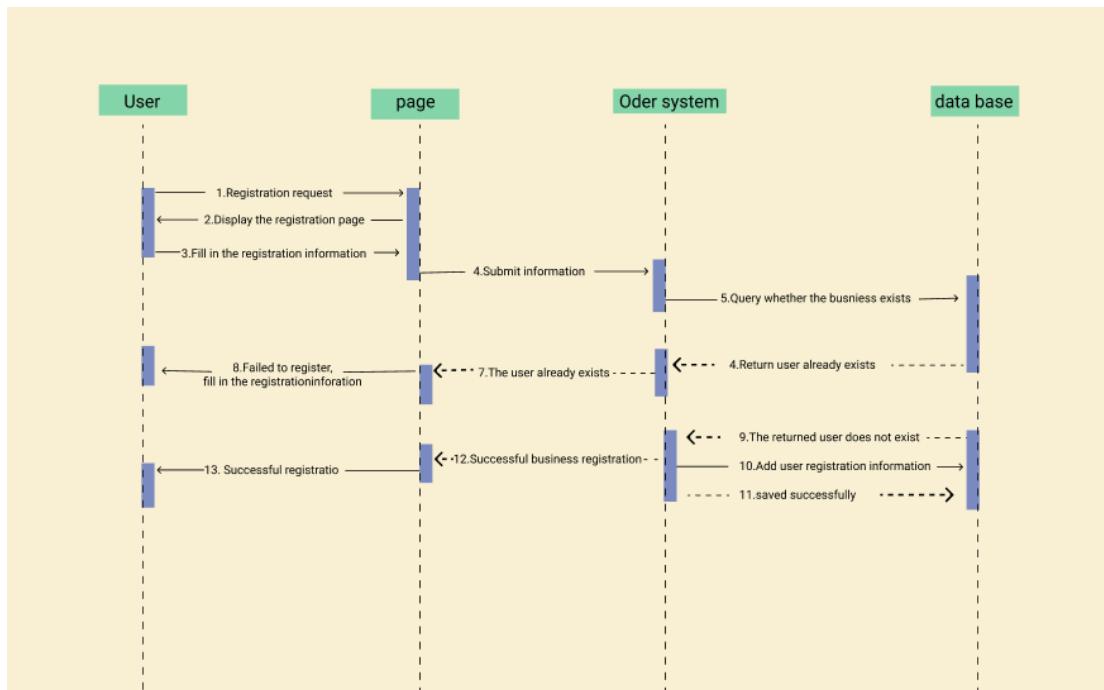


User refund sequence diagram:

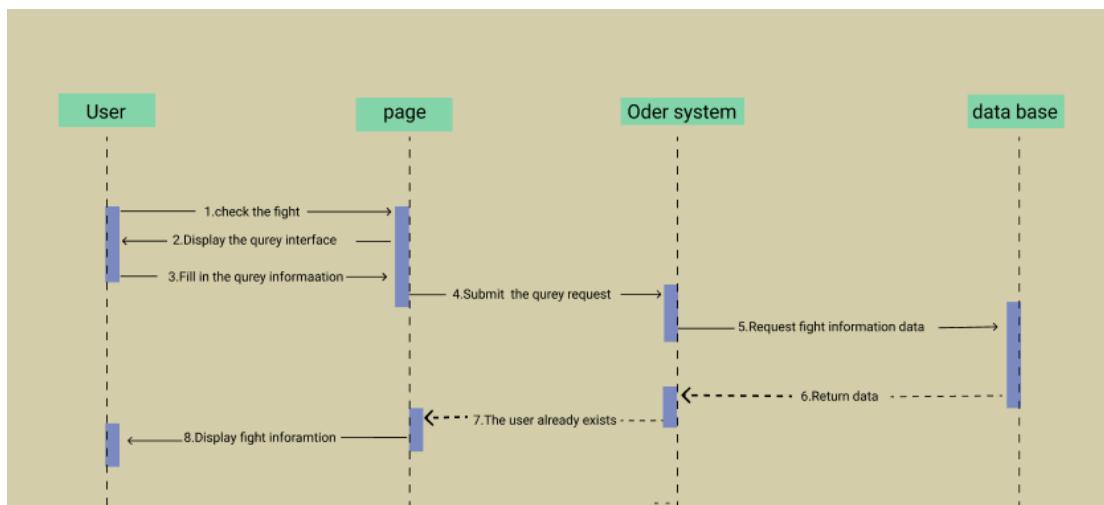


3.3.2 Two-layer C / S structure

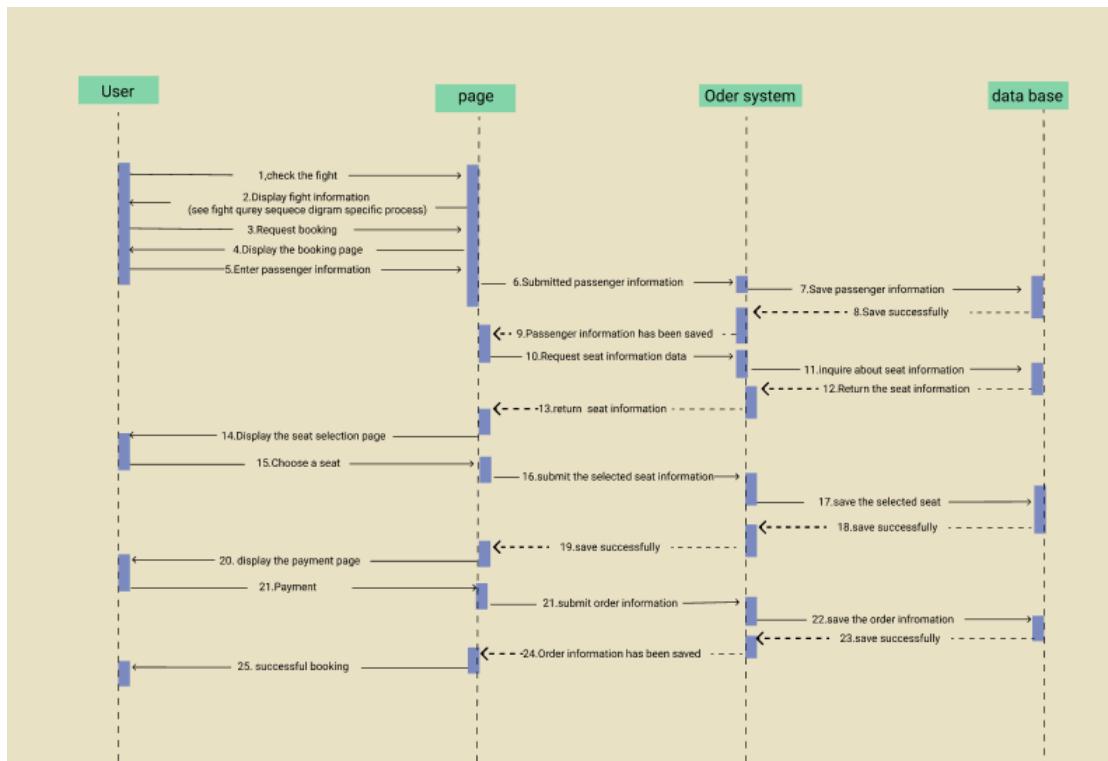
User registration timing chart:



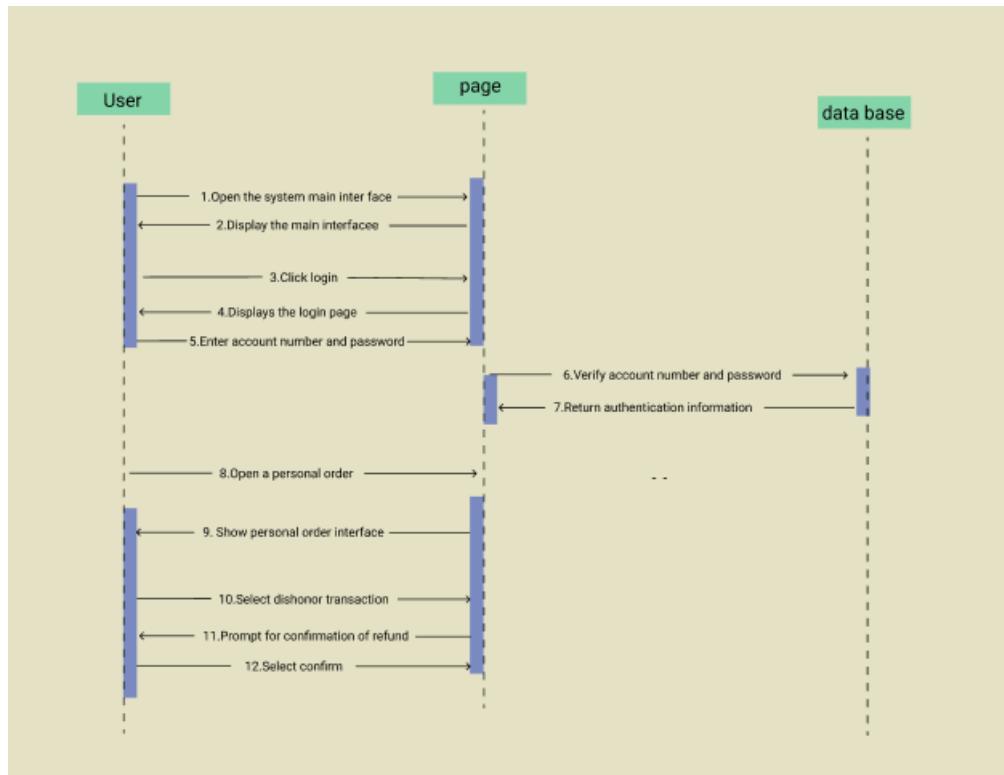
Fight inquiry sequence diagram:



User booking sequence diagram:



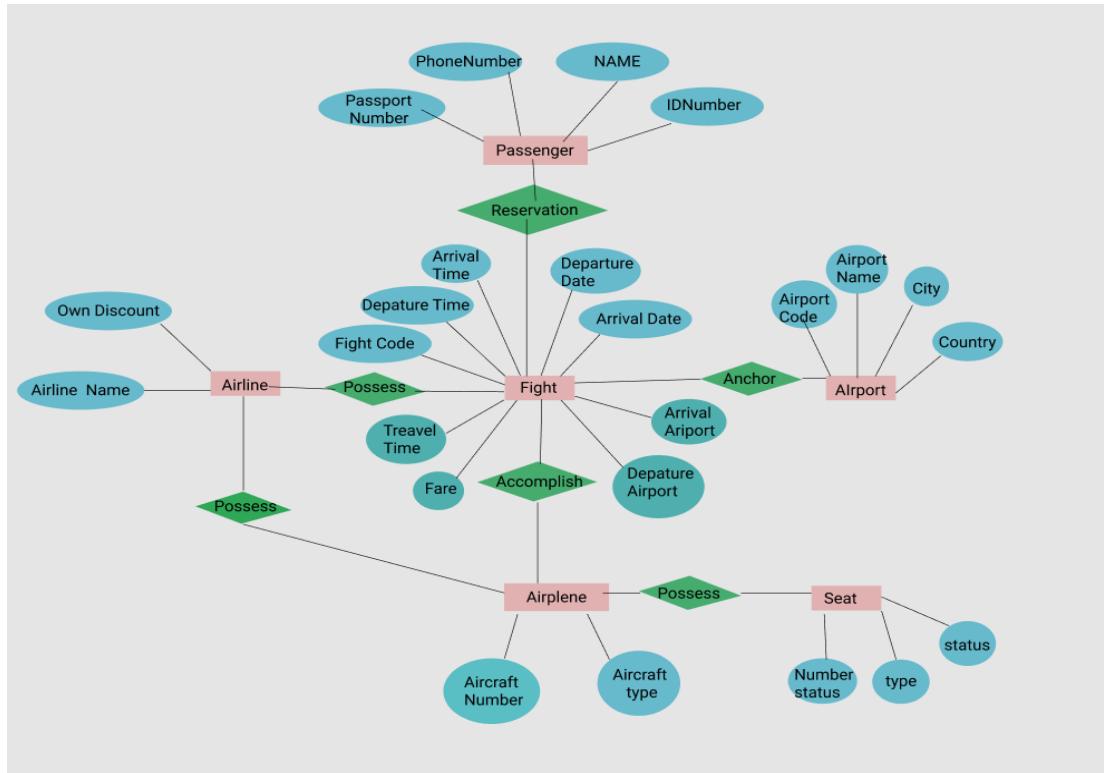
User refund sequence diagram:



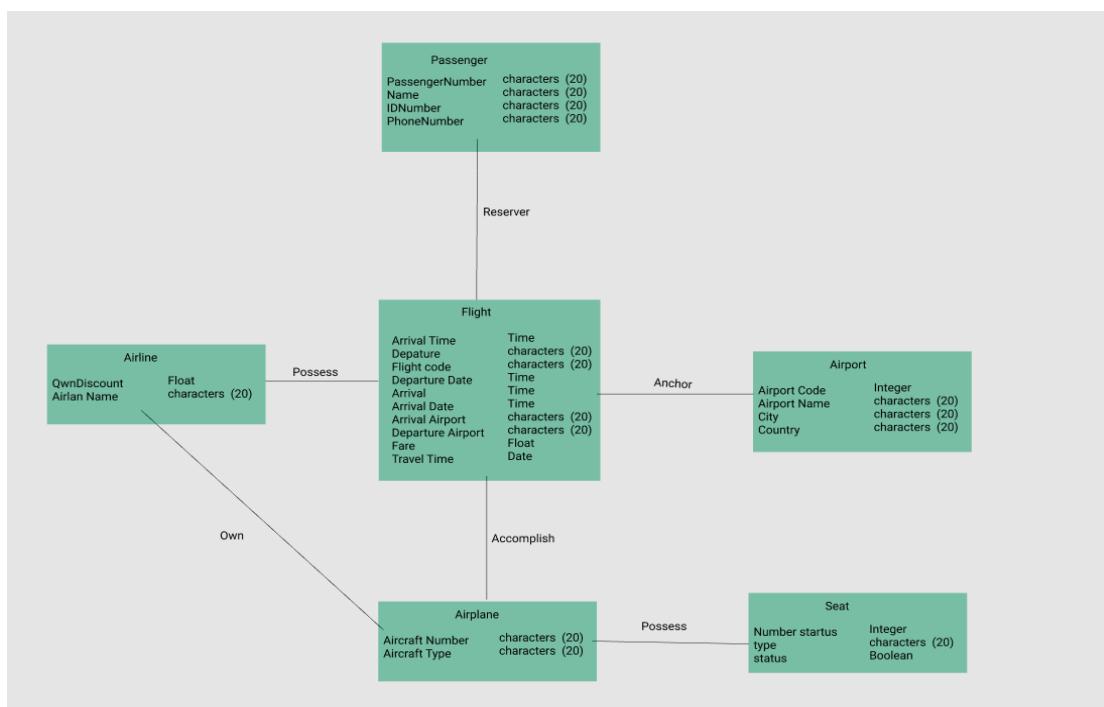
4. Database Design

4.1 Conceptual design (E-R diagram)

E-R Diagram:



Conceptual structure of design:



4.2 Logical structure design

The ER graph is transformed into a relational schema, and the attributes, functional dependencies, candidate codes and external codes of each relationship are determined, and the relational schema is normalized to at least 3NF:

Passengers (passport number, name, ID number, mobile phone number)

Reservation (passport number, flight number)

Flight (flight number, company name, aircraft number, departure date, departure time, arrival date, arrival time, departure airport, destination airport, travel time, cost)

Airlines (company name, discount)

Parking (airport number, flight number)

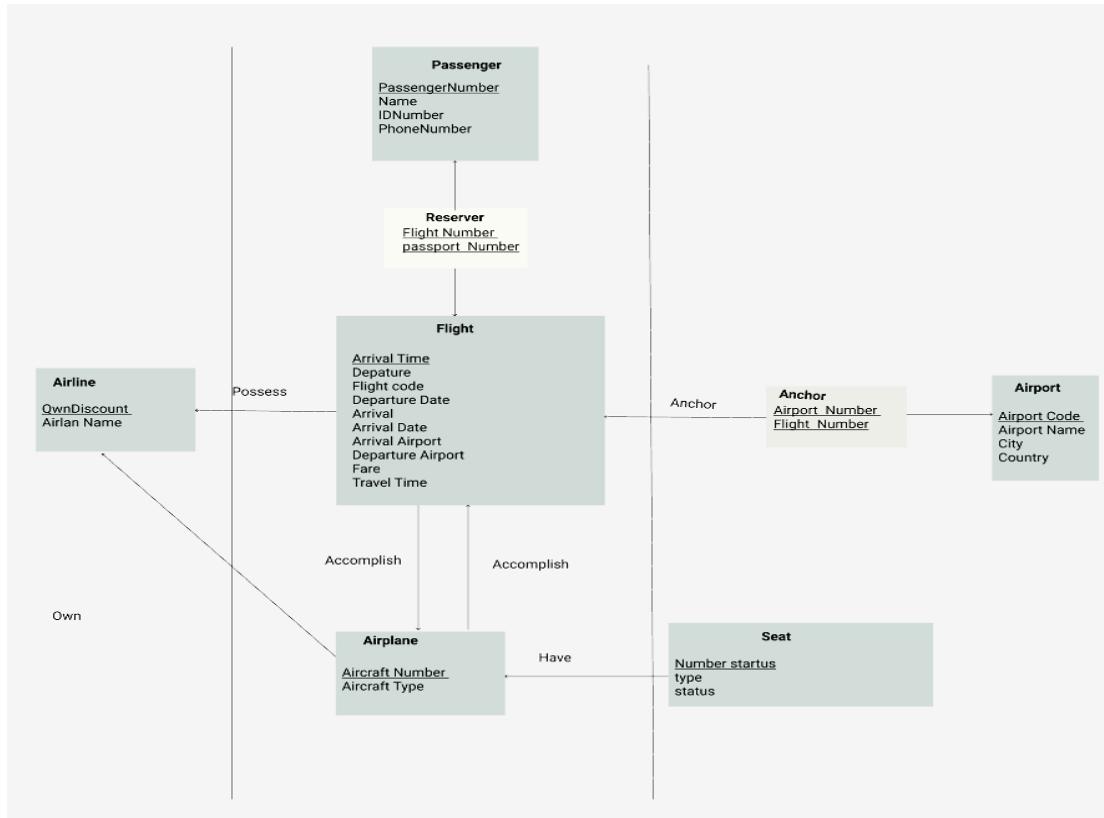
Airport (airport number, airport name, city, country)

Aircraft (aircraft number, flight number, company name, aircraft type)

Seat (seat number, aircraft number, type, status)

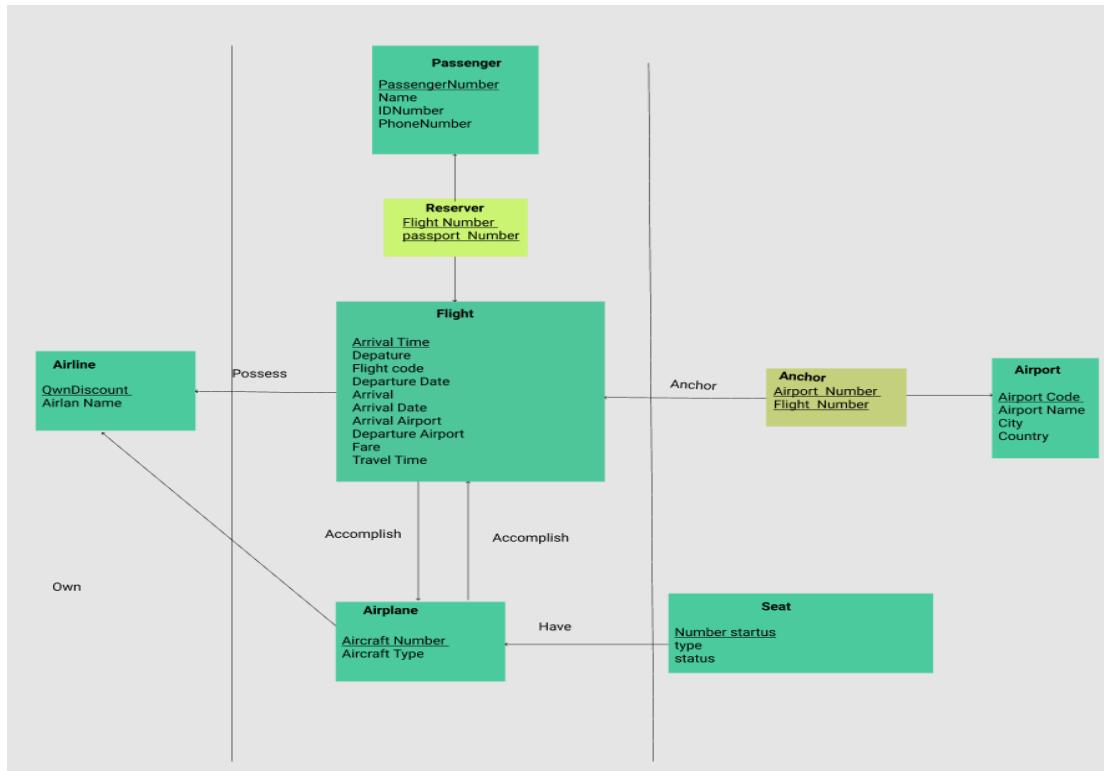
In the above relational mode, the underline is the main code of horizontal line, and the underline is the outer code of wavy line

Logical structure diagram of design:



4.3 Physical structure design

Physical structure diagram of design:



5. Database Implementation and Operation

5.1 Introduction of database management system

Starting from the database design, the database is designed according to the data involved in the system and their relations and constraints. Then, according to the basic access operation of the database, the basic Dao is designed for the later complex functions. According to the requirements, we divide the function into three main parts: vote, reservation and refund. Based on this, the front-end page is designed, and then according to the data required by the front-end page, the servlet is used to process the data retrieved from the database and the data returned from the front-end page to realize the interaction between the front-end and the front-end.

The main design idea of front-end page is simple and clear, and then unify the distribution and style of each page. The front page is responsible for display and response. The back-end designs the servlet program according to the three use case diagrams of ballot, booking and refund. It stores the flight information, passenger information and seat selection information selected by the user into the session domain. Once the transaction submission fails, the data is rolled back, and the user's temporary seat selection number is displayed back to the seat selection page with the servlet. When the user clicks on the ticket reservation, the user's information will be preserved when the user returns the ticket, the

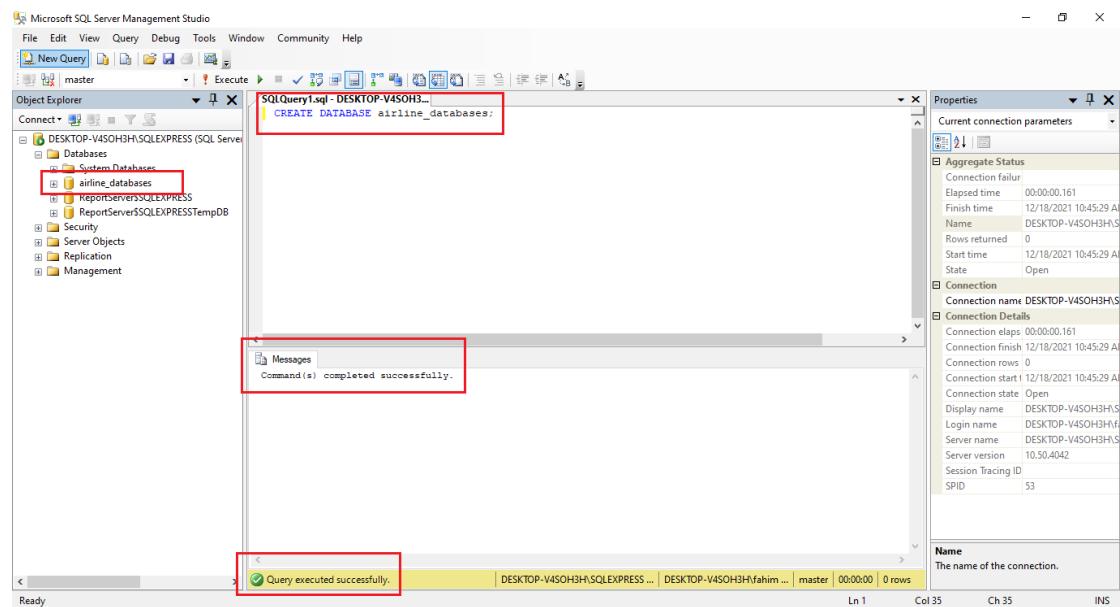
booking information of the user in the database will be deleted and the session field will be cleared. The data used by the servlet is obtained from the database by Dao, and the data processed by the servlet is also stored in the database by Dao, so as to realize the information interaction between the front-end, business processing and database layers.

5.2 Database creation and data entry

Database creation, table creation and data entry code screenshot

Database Creation:

[CREATE DATABASE](#) airline_databases



USE airline_databases

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'DESKTOP-V4SOH3H\SQLEXPRESS' is selected, with 'airline_databases' chosen under 'Databases'. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3H\ahim (3)' contains the command: 'USE airline_databases;'. Below the command, the 'Messages' pane displays the message: 'Command(s) completed successfully.' A red box highlights this message. At the bottom of the screen, a status bar shows 'Query executed successfully.' A red box also highlights this status message.

```
USE airline_databases;
```

Messages
Command(s) completed successfully.

Query executed successfully.

Table Creation:

Airline:

```
CREATE TABLE [dbo].[Airline] (
    [Owndiscount] [float] NOT NULL ,
    [AirlineName] [char] (40) NULL,
)ON [PRIMARY]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, a connection to 'DESKTOP-V4SOH3H\SQLEXPRESS' is selected, with 'Tables' chosen under 'airline_databases'. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3H\ahim (3)' contains the 'CREATE TABLE' command for 'Airline'. A red box highlights the table definition. Below the command, the 'Messages' pane displays the message: 'Command(s) completed successfully.' A red box highlights this message. At the bottom of the screen, a status bar shows 'Query executed successfully.' A red box also highlights this status message.

```
CREATE TABLE [dbo].[Airline] (
    [Owendiscount] [float] NOT NULL ,
    [AirlineName] [char] (40) NULL,
)ON [PRIMARY]
```

Messages
Command(s) completed successfully.

Query executed successfully.

Airplane:

```
CREATE TABLE [dbo].[Airplane] (
[AircraftNumber] [character](20) NOT NULL ,
[AircraftType] [character] (20) NULL,
)ON [PRIMARY]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'airline_databases' database, the 'Tables' node is expanded, showing 'dbo.Airline' and 'dbo.Airplane'. The 'dbo.Airplane' node is highlighted with a red box. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3HVahim (53)' contains the CREATE TABLE script. The output pane at the bottom shows a green checkmark icon and the message 'Query executed successfully.' A red box highlights this message. On the right, the Properties window displays connection details for the current session.

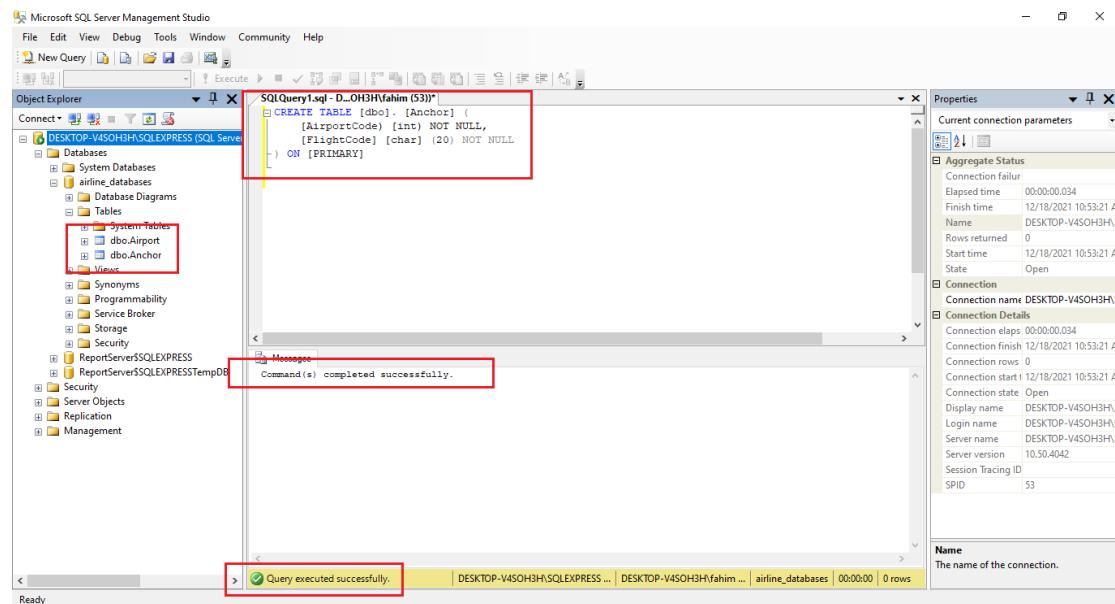
Airport:

```
CREATE TABLE [dbo].[Airport](
[AirportCode][int] NOT NULL,
[City] [char] (20) NULL,
[Country] [char] (20) NULL,
[FlightCode] [char] (20) NULL
)ON [PRIMARY]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'airline_databases' database, the 'Tables' node is expanded, showing 'dbo.System Tables' and 'dbo.Airport'. The 'dbo.Airport' node is highlighted with a red box. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3HVahim (53)' contains the CREATE TABLE script. The output pane at the bottom shows a green checkmark icon and the message 'Query executed successfully.' A red box highlights this message. On the right, the Properties window displays connection details for the current session.

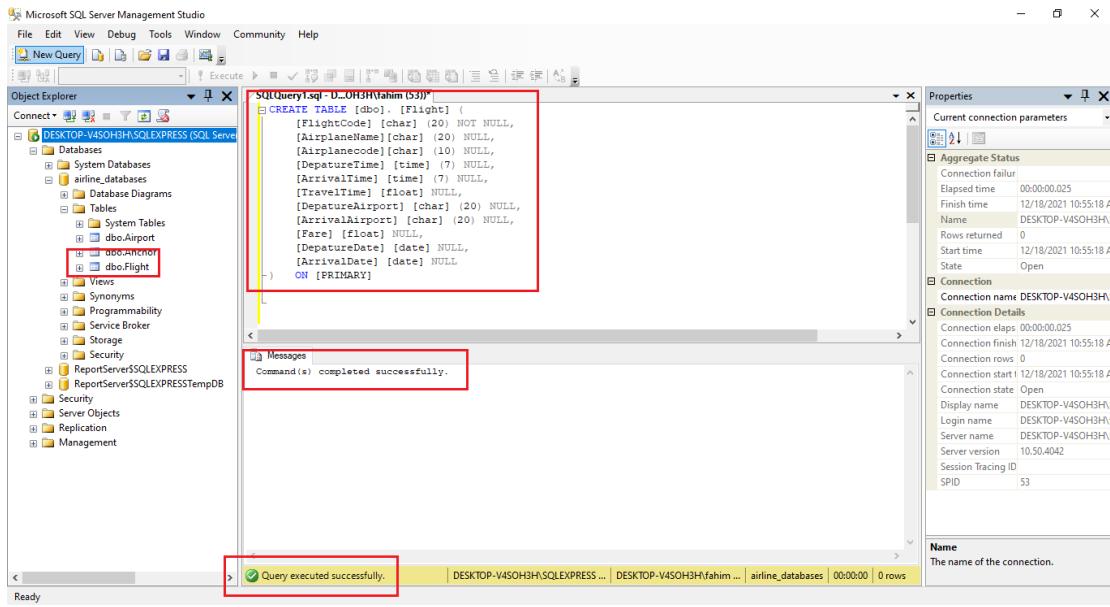
Anchor:

```
CREATE TABLE [dbo].[Anchor] (
    [AirportCode] [int] NOT NULL,
    [FlightCode] [char] (20) NOT NULL
) ON [PRIMARY]
```



Flight:

```
CREATE TABLE [dbo].[Flight] (
    [FlightCode] [char] (20) NOT NULL,
    [AirplaneName] [char] (20) NULL,
    [Airplanecode] [char] (10) NULL,
    [DepartureTime] [time] (7) NULL,
    [ArrivalTime] [time] (7) NULL,
    [TravelTime] [float] NULL,
    [DepartureAirport] [char] (20) NULL,
    [ArrivalAirport] [char] (20) NULL,
    [Fare] [float] NULL,
    [DepartureDate] [date] NULL,
    [ArrivalDate] [date] NULL
) ON [PRIMARY]
```



Order:

```

CREATE TABLE [dbo].[Order] (
    [orderNumber] [nchar] (20) NOT NULL,
    [startCity] [nchar] (20) NULL,
    [endCity] [nchar] (20) NULL,
    [price] [float] NULL,
    [startAirport] [nchar] (20) NULL,
    [endAirport][nchar] (20) NULL,
    [startDace] [date] NULL,
    [startTime] [time] (7) NULL,
    [endDate] [date] NULL,
    [endTime] [time] (7) NULL,
    [seatType] [nchar] (20) NULL,
    CONSTRAINT [PK_Order] PRIMARY KEY CLUSTERED
    (
        [orderNumber] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
)ON [PRIMARY]

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'airline_databases' database, the 'Tables' node is expanded, and the 'dbo.Order' table is selected. The 'SQLQuery1.sql - D... (53)' tab in the main window contains the following SQL code:

```

CREATE TABLE [dbo].[Order] (
    [orderNumber] [nchar] (20) NOT NULL,
    [startCity] [nchar] (20) NULL,
    [endCity] [nchar] (20) NULL,
    [price] [float] NULL,
    [startAirport] [nchar] (20) NULL,
    [endAirport] [nchar] (20) NULL,
    [startDate] [date] NULL,
    [startTime] [time] (7) NULL,
    [endDate] [date] NULL,
    [endTime] [time] (7) NULL,
    [seatType] [nchar] (20) NULL,
    CONSTRAINT [FK_Order] PRIMARY KEY CLUSTERED
    (
        [orderNumber] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON) ON [PRIMARY]
)

```

The 'Messages' pane at the bottom right shows the message "Command(s) completed successfully." The status bar at the bottom indicates "Query executed successfully." and "0 rows".

Passenger:

```

CREATE TABLE [dbo].[Passenger] (
    [PassportNumber] [char] (20) NOT NULL ,
    [Name] [char] (10) NULL,
    [IDNumber] [char] (18) NULL ,
    [PhoneNumber] [char] (20) NULL
)ON [PRIMARY]

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, under the 'airline_databases' database, the 'Tables' node is expanded, and the 'dbo.Passenger' table is selected. The 'SQLQuery1.sql - D... (53)' tab in the main window contains the following SQL code:

```

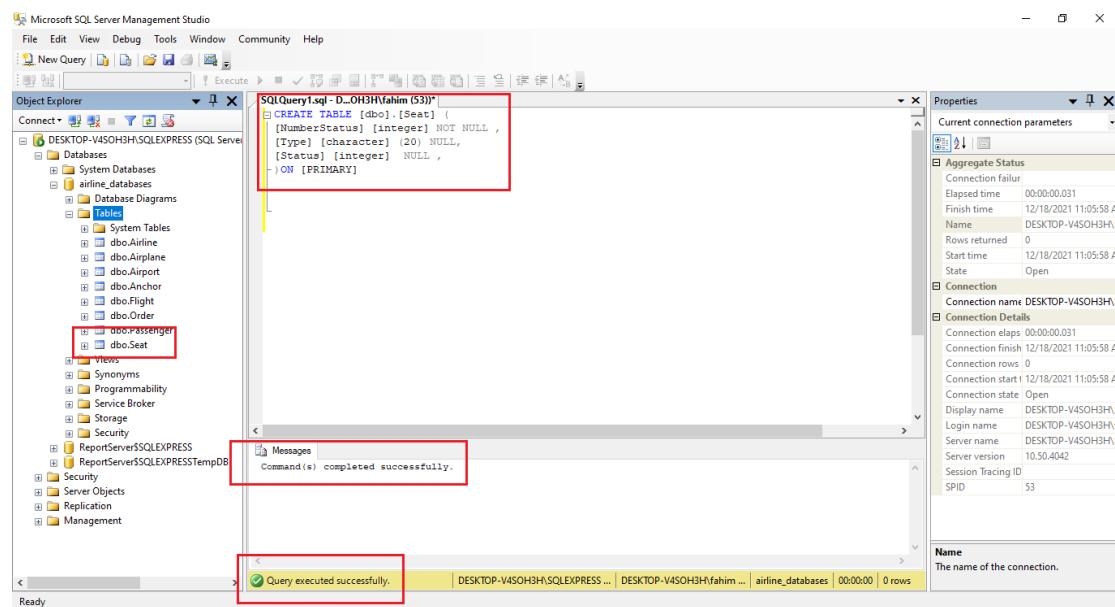
CREATE TABLE [dbo].[Passenger] (
    [PassportNumber] [char] (20) NOT NULL ,
    [Name] [char] (10) NULL,
    [IDNumber] [char] (18) NULL ,
    [PhoneNumber] [char] (20) NULL
)ON [PRIMARY]

```

The 'Messages' pane at the bottom right shows the message "Command(s) completed successfully." The status bar at the bottom indicates "Query executed successfully." and "0 rows".

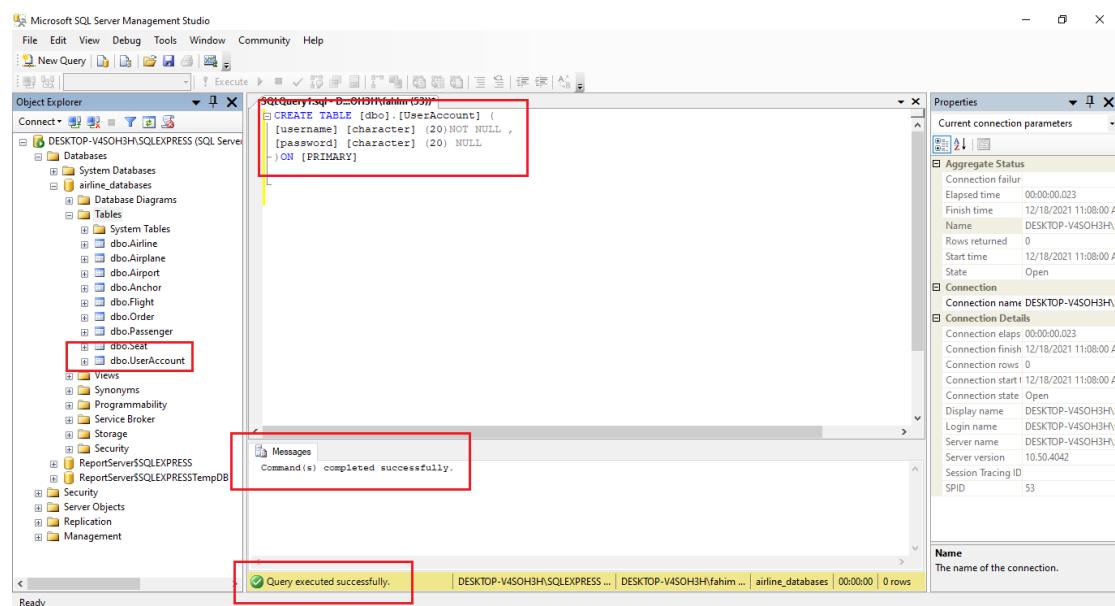
Seat:

```
CREATE TABLE [dbo].[Seat] (
[NumberStatus] [integer] NOT NULL ,
[Type] [character] (20) NULL,
[Status] [integer] NULL ,
)ON [PRIMARY]
```



User:

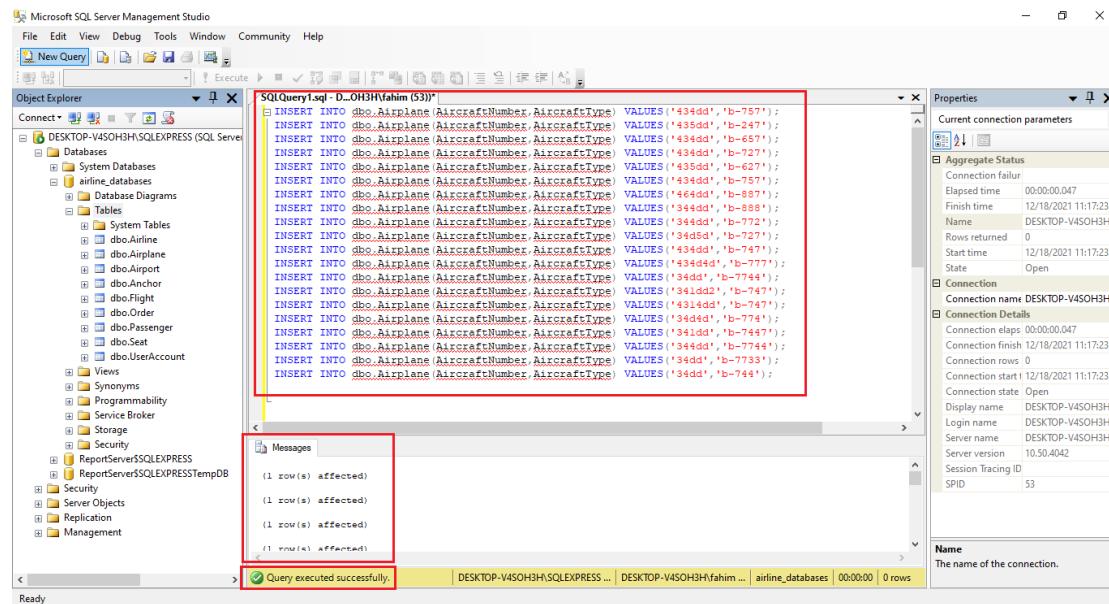
```
CREATE TABLE [dbo].[UserAccount] (
[username] [character] (20)NOT NULL ,
[password] [character] (20) NULL
)ON [PRIMARY]
```



Data Entry:

Data insert at airplane table:

```
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434dd','b-757');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('435dd','b-247');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434dd','b-657');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434dd','b-727');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('435dd','b-627');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434dd','b-757');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('464dd','b-887');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('344dd','b-888');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('344dd','b-772');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('34d5d','b-727');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434dd','b-747');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('434d4d','b-777');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('34dd','b-7744');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('341dd2','b-747');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('4314dd','b-747');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('34d4d','b-774');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('341dd','b-7447');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('344dd','b-7744');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('34dd','b-7733');
INSERT INTO dbo.Airplane(AircraftNumber,AircraftType) VALUES('34dd','b-744');
```



Data insert at Airport table:

```
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES
('58450','dhaka','bangladesh','dhsbgf');
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES
('44499','dhaka','bangladesh','jakdek');
```

```
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('58540','dhaka','bangladesh','dhsbgf');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('47894','dhaka','bangladesh','jausgi');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('78493','dhaka','bangladesh','rgeghi');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('98495','dhaka','bangladesh','werssg');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('78742','dhaka','bangladesh','xtrdyh');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('35776','dhaka','bangladesh','qgfrnb');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('25497','dhaka','bangladesh','ndrtsg');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('98495','dhaka','bangladesh','pouidg');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('35468','dhaka','bangladesh','asxgre');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('95745','dhaka','bangladesh','veszra');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('96264','dhaka','bangladesh','hdrdrt');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('76499','dhaka','bangladesh','qgbtyh');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('98764','dhaka','bangladesh','mtcfyu');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('78652','dhaka','bangladesh','xcvgrd');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('35445','dhaka','bangladesh','adfvvb');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('46786','dhaka','bangladesh','btrhsa');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('24578','dhaka','bangladesh','xcvxcf');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('15273','dhaka','bangladesh','rtbgnf');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('52045','dhaka','bangladesh','caewtt');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('35240','dhaka','bangladesh','pytfvg');  
INSERT INTO dbo.Airport(AirportCode,City,Country,FlightCode) VALUES  
('87820','dhaka','bangladesh','xzcvsd');
```

```

INSERT INTO dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate) VALUES ('FFES','China Southern','44ddw','10:34:09 AM','7:04:09 AM','3.2','Bejing','Wuhan','342.3','1-28-2021','1-28-2018');

INSERT INTO dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate) VALUES ('FFES','Eastern Airlines','46tgc','6:3:09 AM','9:04:09 PM','5.6','Wuhan','Guangzhou','342.3','1-28-2021','1-28-2021');

INSERT INTO dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate) VALUES ('KLHL','China Southern','44ddw','10:34:09 PM','7:04:09 AM','1.5','Wuhan','Beijing','342.3','6-2-2021','6-28-2021');

INSERT INTO dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate) VALUES ('MUUI','MALINDO Airline','96dsg','3:34:09 PM','5:04:04 AM','6.5','Wuhan','Mumbai','862.3','2-28-2021','2-28-2021');

```

Data insert at Flight table:

INSERT INTO

```

dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)
VALUES ('FFES','China Southern','44ddw','10:34:09 AM','7:04:09 AM','3.2','Bejing','Wuhan','342.3','1-28-2021','1-28-2018');

```

INSERT INTO

```

dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)
VALUES ('FFES','Eastern Airlines','46tgc','6:3:09 AM','9:04:09 PM','5.6','Wuhan','Guangzhou','342.3','1-28-2021','1-28-2021');

```

INSERT INTO

```

dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)
VALUES ('KLHL','China Southern','44ddw','10:34:09 PM','7:04:09 AM','1.5','Wuhan','Beijing','342.3','6-2-2021','6-28-2021');

```

INSERT INTO

```

dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)
VALUES ('MUUI','MALINDO Airline','96dsg','3:34:09 PM','5:04:04 AM','6.5','Wuhan','Mumbai','862.3','2-28-2021','2-28-2021');

```

INSERT INTO

```

dbo.Flight(FlightCode,AirplaneName,AirplaneCode,DepatureTime,ArrivalTime,TravelTime,DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)

```

```
VALUES ('KHFG','KD Airline','16tjh','7:34:09 PM','1:04:09  
PM','6.2','Wuhan','Lasa','342.3','8-28-2021','8-28-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('DFWD','Native Thai','152gf','10:34:09 AM','12:01:09  
AM','3.2','Wuhan','Kunming','342.3','5-8-2021','5-8-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('SGGV','China Southern','479DER','10:34:09 AM','3:02:09  
PM','3.2','Wuhan','Pakistan','342.3','1-28-2021','1-28-2018');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('JGKBN','Scot Airline','496dw','10:34:09 AM','2:04:01  
AM','3.2','Wuhan','Bali','342.3','1-28-2021','1-28-2018');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('KJHKD','Air Machigan','46mc','6:34:02 PM','9:04:09  
AM','3.2','Wuhan','Washington','4569.3','6-9-2021','6-9-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('SDGRF','Air Beijing','BJ454','1:34:09 PM','10:04:09  
PM','3.2','Wuhan','Beijing','342.3','5-12-2021','5-12-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('DSRGFD','US Bangla','Boeng178','1:00:00 AM','3:00:00  
AM','3.2','Wuhan','Dhaka','3752.3','1-28-2021','1-28-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('FFES','China Southern','44ddw','10:34:09 AM','10:04:09  
AM','3.2','Wuhan','Shanghai','342.3','5-8-2021','5-8-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('DFWD','NAtive Thai','152gf','10:34:09 AM','7:04:09  
AM','3.2','Wuhan','Kunming','342.3','5-8-2021','5-8-2021')
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('FFES','China Southern','44ddw','10:34:09 AM','7:04:09  
AM','3.2','Wuhan','Kalkatta','342.3','12-28-2021','12-28-2021');
```

INSERT INTO

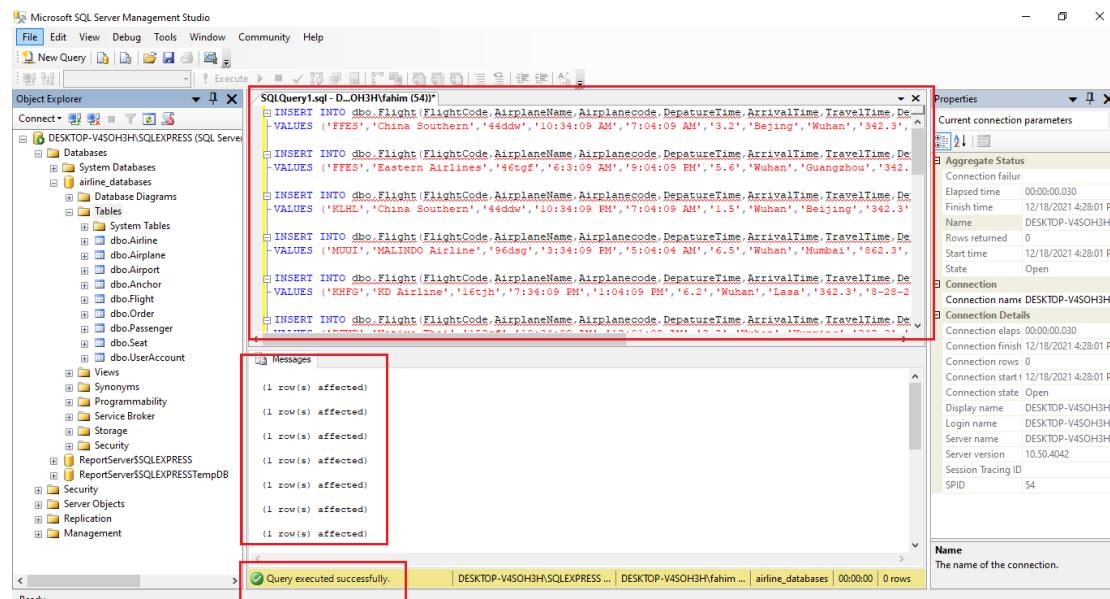
```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('FFES','China Eastern','644kl','10:34:09 AM','7:04:09  
AM','3.2','Wuhan','Lanzhou','342.3','1-28-2021','1-28-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('FFES','Air Asia','44ddw','10:34:09 AM','7:04:09  
AM','3.2','Wuhan','Giangxi','342.3','12-2-2021','12-2-2021');
```

INSERT INTO

```
dbo.Flight(FlightCode,AirplaneName,Airplanecode,DepatureTime,ArrivalTime,TravelTime,  
DepatureAirport,ArrivalAirport,Fare,DepatureDate,ArrivalDate)  
VALUES ('FFES','China Southern','44ddw','10:34:09 AM','7:04:09  
AM','3.2','Bejing','Wuhan','342.3','12-28-2021','12-28-2021');
```



Data insert for Passenger table:

```
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('E-32223','ZAMAN','493882NJD','5465335');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('T-4245','MANIK','493882NJD','5658375');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('S45W62','TOMOY','493882NJD','57643');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('C-4567','MAMUN','493882NJD','96876553');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('H-553G','PIAS','493882NJD','36756453');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('CE5B35','ZUBAYER','493882NJD','3865645');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('E-53B55','SARKAR','493882NJD','8675453');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('AD56B53','KHAN','493882NJD','675646756');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('E-56563','ZAHIN','493882NJD','87564556');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('T-5645','BRTAR','493882NJD','678675436');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('EAD6G3G','MADHARCHUD','493882NJD','47865746');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('ET-87785','MAGIBETI','493882NJD','5768786');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('CIN-NM3','SCHODI','493882NJD','86756466');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('AS-5356B','IEKSSS','493882NJD','7564567');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('INE-BVB','EMON','493882NJD','8675646735');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('EA-3355','NAIKHAN','493882NJD','857645365');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('EE-3VBN','AMIN KHAN','493882NJD','646537876');  
INSERT INTO dbo.Passenger(PassportNumber,Name,IDNumber,PhoneNumber) VALUES  
('EAA36GV','JAMIN KHAN','493882NJD','8675645656');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery1.sql - D_OH3HVahim (54)' is open, displaying a large block of SQL code for inserting data into the 'dbo.Passenger' table. The code consists of 28 INSERT statements, each adding a new passenger record with details like name, ID number, and phone number. Below the code, the 'Messages' pane shows 10 rows affected for each insert statement. At the bottom of the window, a green message box indicates 'Query executed successfully.' The status bar at the bottom right shows 'Ln 19 Col 1 Ch 1 INS'. To the right of the main window, the 'Properties' pane displays connection details for the current session.

```

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('E-32223', 'ZAHAN', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('T-4245', 'MONIK', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('S45862', 'TOMOY', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('C-4567', 'MAMUN', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('H-553G', 'FIAS', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('CE5B35', 'ZUBAYER', '4', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('E-53B5', 'SARKAR', '4', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('F-56563', 'ZAHIN', '49', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('AD64B53', 'KHAN', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('T-5645', 'BRTAR', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('EAD63G', 'MAHDARCHUD', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('ET-87785', 'MAGIBETI', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('CIN-NM3', 'SCHODI', '4', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('AS-5356B', 'IEKSSS', '4', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('INE-BVB', 'ENON', '493', '0987654321')
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber) VALUES ('EA-3355', 'NAIKHAN', '493', '0987654321')

```

Data insert for Seat table:

```

INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('1','BUSNIESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('2','BUSNIESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('3','BUSNIESS-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('4','BUSNIESS-CLASS','3');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('5','BUSNIESS-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('6','BUSNIESS-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('7','BUSNIESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('10','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('11','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('12','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('13','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('14','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('15','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('16','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('17','ECONOMY-CLASS','3');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('18','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('19','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('20','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('21','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('22','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('23','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('24','ECONOMY-CLASS','3');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('25','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('26','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('27','ECONOMY-CLASS','3');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('28','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('29','ECONOMY-CLASS','1');

```

```
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('30','ECONOMY-CLASS','2');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the airline_databases database and its tables like System Tables, dbo.Airline, dbo.Seat, etc. The central pane displays the SQL query:

```
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('30','ECONOMY-CLASS','2');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('11','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('12','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('13','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('14','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('15','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('16','BUSINESS-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('17','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('18','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('19','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('20','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('21','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('22','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('23','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('24','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('25','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('26','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('27','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('28','ECONOMY-CLASS','1');
INSERT INTO dbo.Seat(NumberStatus,Type,Status) VALUES ('29','ECONOMY-CLASS','1');
```

The results pane shows the output of the query:

```
(1 row(s) affected)
```

A message at the bottom indicates "Query executed successfully." The Properties pane on the right shows connection details.

Data insert for Airline table:

```
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('3','China Southern Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('1','American Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('3','Delta Air Lines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('22','Frontier Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('4','Hawaiian Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('10','Air China');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('3','China Eastern Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('3','Hainan Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('21','Shenzhen Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('12','Sichuan Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('1','XiamenAir');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('00','Jetstar Asia Airways');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('3','Scoot');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('13','SilkAir');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('5','Singapore Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('6','Biman Bangladesh Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('9','US-Bangla Airlines');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('11','Regent Airways');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('8','Novoair');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('2','IndiGo');
INSERT INTO dbo.Airline(OwnDiscount,AirlineName) VALUES ('5','GoAir');
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including System Databases, System Tables, and Tables like dbo.Airlines, dbo.Airplane, etc. The central pane displays a query window titled 'SQLQuery1.sql - D... (54)' containing the following SQL code:

```

INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('3','China Eastern Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('3','Hainan Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('21','Shenzhen Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('11','XiamenAir');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('13','Scoot');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('5','Singapore Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('6','Biman Bangladesh Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('9','US-Bangla Airlines');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('11','Regent Airways');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('8','Novoair');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('2','IndiGo');
INSERT INTO dbo.Airline_OwnDiscount(AirlineName) VALUES ('5','GoAir');

```

The 'Messages' pane below the query results shows seven rows affected by the insert operation. The status bar at the bottom indicates 'Query executed successfully.'

Data insert for UserAccount table:

In here I face some problem, its “call string or binary would be truncated.”

Then I overcome this problem by using proper size of string. Its data insert for a passenger user account information. When a passenger come and use my website/application for search or query she/he should create account/login. So we need to store client user name and password.

```

INSERT INTO dbo.UserAccount(username ,password) VALUES('zubayer','3b56v');
INSERT INTO dbo.UserAccount(username ,password) VALUES('amitumi','64363b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('misha','356bn4');
INSERT INTO dbo.UserAccount(username ,password) VALUES('sony','5b6575456');
INSERT INTO dbo.UserAccount(username ,password) VALUES('sayem','b63b64b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('nasim','4b566b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('shihab','4b3v476');
INSERT INTO dbo.UserAccount(username ,password) VALUES('nahim','67nb547n6b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('habib','64b7g57');
INSERT INTO dbo.UserAccount(username ,password) VALUES('zaman','nkies223');
INSERT INTO dbo.UserAccount(username ,password) VALUES('rana','hsegv4');
INSERT INTO dbo.UserAccount(username ,password) VALUES('zahin','vv3y4by5');
INSERT INTO dbo.UserAccount(username ,password) VALUES('manik','5b3673b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('oias','3b56bg65');
INSERT INTO dbo.UserAccount(username ,password) VALUES('pias','3b65bb5b');
INSERT INTO dbo.UserAccount(username ,password) VALUES('amin','b36v56b');

```

```

INSERT INTO dbo.UserAccount(username ,password) VALUES('tuhin','35bv65');
INSERT INTO dbo.UserAccount(username ,password) VALUES('kamal','5n7b6n');
INSERT INTO dbo.UserAccount(username ,password) VALUES('laoshi','b35b76');
INSERT INTO dbo.UserAccount(username ,password) VALUES('santi','b453636v');
INSERT INTO dbo.UserAccount(username ,password) VALUES('lalamia','3nb6bn');
INSERT INTO dbo.UserAccount(username ,password) VALUES('fahim','fm6fm6');
INSERT INTO dbo.UserAccount(username ,password) VALUES('sayma','sm03f');

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'airline_databases' database is selected. In the center, a query window titled 'SQLQuery1.sql - D... (54)' contains several INSERT statements into the 'dbo.UserAccount' table. The results pane shows '21 rows affected'. The status bar at the bottom indicates 'Query executed successfully.' and '21 rows'.

5.3 Database query

Airline:

```

SELECT [OwnDiscount],[AirlineName]
FROM [airline_databases].[dbo].[Airline]

```

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'master' database is selected. In the center, a query window titled 'SQLQuery1.sql - DESKTOP-V45O...' contains a SELECT statement from the 'dbo.Airline' table. The results pane shows 21 rows of data, including entries for China Southern Airlines, American Airlines, Delta Air Lines, Frontier Airlines, Hawaiian Airlines, Air China, China Eastern Airlines, Hainan Airlines, Shenzhen Airlines, Sichuan Airlines, XiamenAir, Jetstar Asia Airways, Scoot, SilkAir, Singapore Airlines, Biman Bangladesh A..., US-Bangla Airlines, Regent Airways, Novair, IndGo, and GoAir. The status bar at the bottom indicates 'Query executed successfully.' and '21 rows'.

Airplane:

```
SELECT [AircraftNumber],[AircraftType]  
FROM [airline_databases].[dbo].[Airplane]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'airline_databases' database which contains the 'dbo.Airplane' table. The Results tab displays the output of the provided SQL query. The Properties window on the right shows connection details like name, server, and session tracing ID.

AircraftNumber	AircraftType
1	b-757
2	b-247
3	b-657
4	b-727
5	b-627
6	b-757
7	b-646d
8	b-888
9	b-772
10	b-727
11	b-747
12	b-777
13	b-7744
14	b-747
15	b-747
16	b-774
17	b-7447
18	b-7744
19	b-7733
20	b-744

Airport:

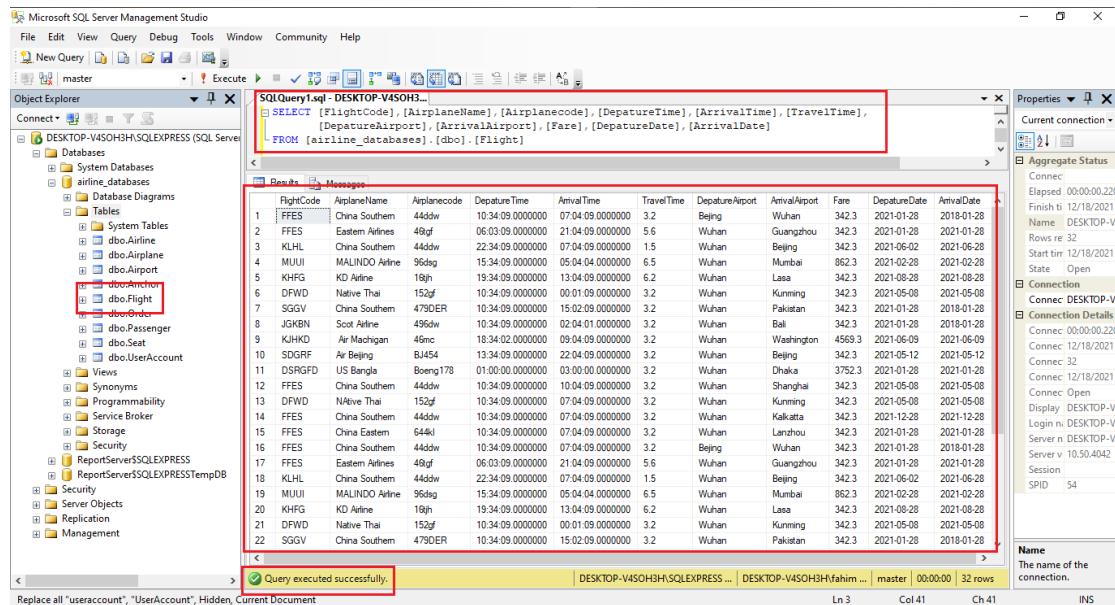
```
SELECT [AirportCode],[City],[Country],[FlightCode]  
FROM [airline_databases].[dbo].[Airport]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'airline_databases' database which contains the 'dbo.Airport' table. The Results tab displays the output of the provided SQL query. The Properties window on the right shows connection details like name, server, and session tracing ID.

AirportCode	City	Country	FlightCode
1	58450	dhaka	bangladesh
2	44499	dhaka	bangladesh
3	58540	dhaka	bangladesh
4	47894	dhaka	bangladesh
5	78493	dhaka	bangladesh
6	98495	dhaka	bangladesh
7	78742	dhaka	bangladesh
8	35776	dhaka	bangladesh
9	25497	dhaka	bangladesh
10	98495	dhaka	bangladesh
11	35468	dhaka	bangladesh
12	95745	dhaka	bangladesh
13	96264	dhaka	bangladesh
14	76499	dhaka	bangladesh
15	98764	dhaka	bangladesh
16	78652	dhaka	bangladesh
17	35445	dhaka	bangladesh
18	45786	dhaka	bangladesh
19	24578	dhaka	bangladesh
20	15273	dhaka	bangladesh
21	52045	dhaka	bangladesh
22	35240	dhaka	bangladesh
23	87820	dhaka	bangladesh

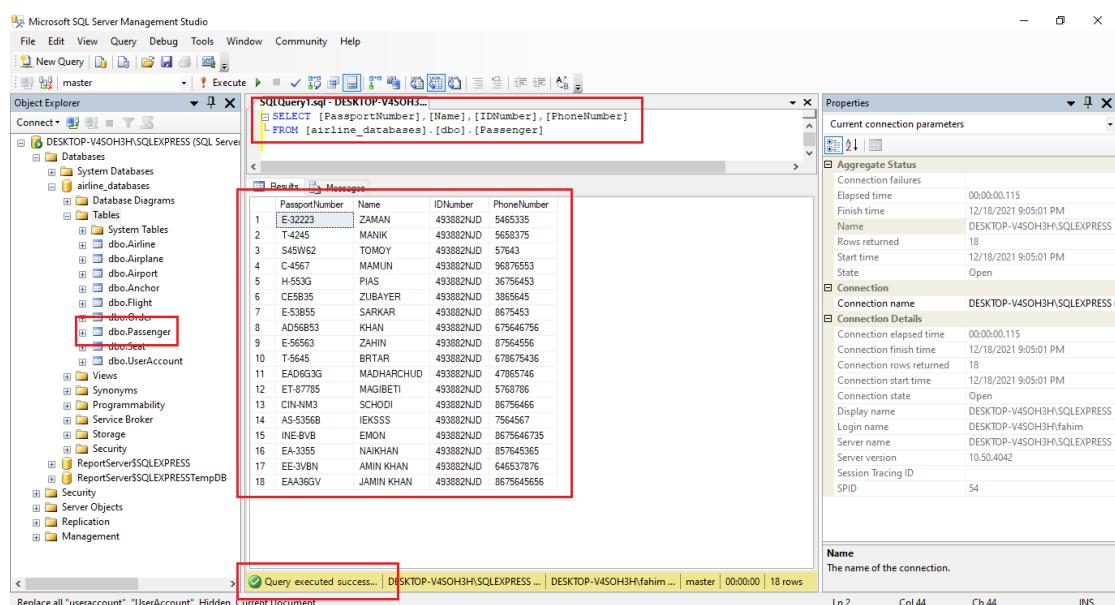
Flight:

```
SELECT [FlightCode].[AirplaneName],[Airplanecode],[DepatureTime],  
       [ArrivalTime],[TravelTime],[DepatureAirport],  
       [ArrivalAirport],[Fare],[DepatureDate],[ArrivalDate]  
FROM [airline_databases].[dbo].[Flight]
```



Passenger:

```
SELECT [PassportNumber],[Name],[IDNumber],[PhoneNumber]  
FROM [airline_databases].[dbo].[Passenger]
```



Seat:

```
SELECT [NumberStatus],[Type],[Status]  
FROM [airline_databases].[dbo].[Seat]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'Tables' node which contains 'Seat'. The central pane displays the results of the executed query:

```
SELECT [NumberStatus],[Type],[Status]  
FROM [airline_databases].[dbo].[Seat]
```

	NumberStatus	Type	Status
1	1	BUSINESS-CLASS	1
2	2	BUSINESS-CLASS	1
3	3	BUSINESS-CLASS	2
4	4	BUSINESS-CLASS	3
5	5	BUSINESS-CLASS	2
6	6	BUSINESS-CLASS	2
7	7	BUSINESS-CLASS	1
8	10	ECONOMY-CLASS	2
9	11	ECONOMY-CLASS	1
10	12	ECONOMY-CLASS	2
11	13	ECONOMY-CLASS	1
12	14	ECONOMY-CLASS	1
13	15	ECONOMY-CLASS	1
14	16	ECONOMY-CLASS	1
15	17	ECONOMY-CLASS	3
16	18	ECONOMY-CLASS	1
17	19	ECONOMY-CLASS	2
18	20	ECONOMY-CLASS	2
19	21	ECONOMY-CLASS	2
20	22	ECONOMY-CLASS	1
21	23	ECONOMY-CLASS	1
22	24	ECONOMY-CLASS	3
23	25	ECONOMY-CLASS	2

The status bar at the bottom indicates "Query executed successfully" and provides connection details.

UserAccount:

```
SELECT [username] ,[password]  
FROM [airline_databases].[dbo].[UserAccount]
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure, including the 'Tables' node which contains 'UserAccount'. The central pane displays the results of the executed query:

```
SELECT [username] ,[password]  
FROM [airline_databases].[dbo].[UserAccount]
```

	username	password
1	aubayer	3t56v
2	amituru	64363b
3	mehna	356614
4	sony	566575456
5	sayem	b53c64b
6	nasrin	4b56b
7	shihab	4b3v476
8	nahin	676b5746b
9	habib	646757
10	zeman	rkies223
11	rana	heevi4
12	zain	vv3y4b5
13	manik	5b3673b
14	oias	3b59bg55
15	piaa	3b65b5b
16	amin	b3v65b
17	tuhin	35v65
18	kamal	5n7b6n
19	laoshi	b35b76
20	santi	b453636v
21	lalania	3rb68n
22	fahim	fm0f6
23	seyma	sm03'

The status bar at the bottom indicates "Query executed successfully" and provides connection details.

5.4 Database control

The software architecture involved in this experiment is two modes of C/S structure: fat client mode and thin client mode, i.e. two-stage structure and three-dimensional structure. The server of fat client mode is only responsible for data management, while the client is responsible for interactive interface and business processing. The thin client joins the application server to deal with the business logic and rules, so that the client's task is only responsible for representation, which simplifies the task.

In this experiment, the aircraft reservation system stores all kinds of data including airport, flight and seat. Business logic includes the corresponding flight, seat, passenger access, and the cost of flight comparison and evaluation to select a better flight. The presentation layer is mainly used to select flights, seats and orders.

If the two-tier C/S structure is selected, the server is only responsible for data management, and the logical processing, including flight comparison, will be carried out directly on the presentation page. That is to use java code to process data in JSP_page, and then display the data on this page. If it is a three-tier C/S structure, we will use the servlet to process the data, and the JSP page will submit the form to the servlet. In terms of security, the two-tier structure directly interacts with the database, which is less secure than the three-tier structure. In terms of

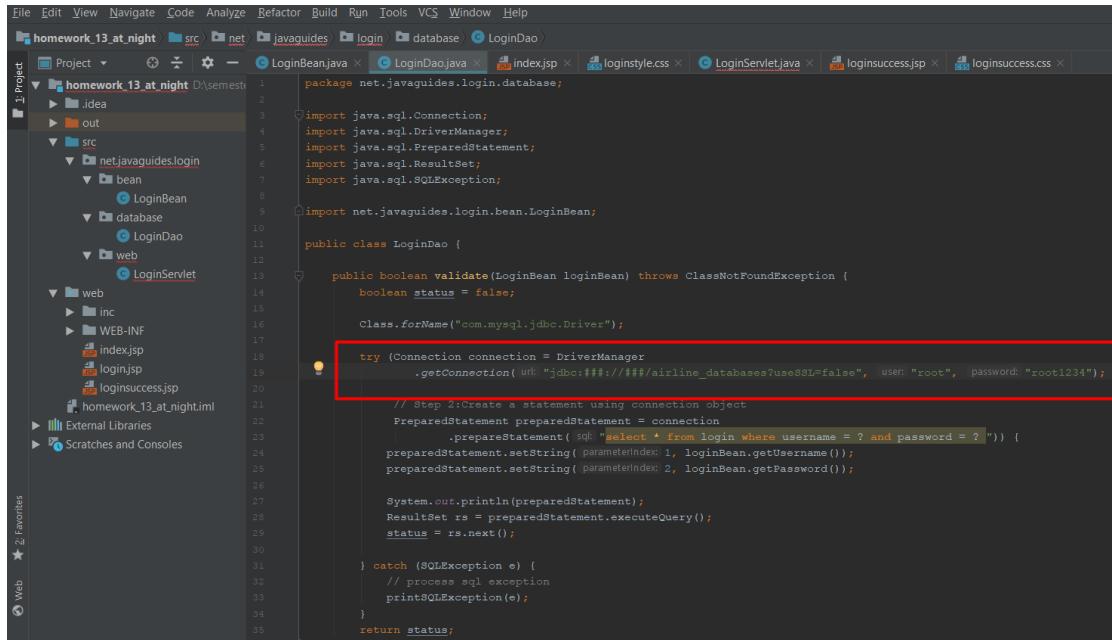
performance, the two-tier client takes on more tasks, and the page loading is slower than the three-tier structure, so the performance is not dominant. Moreover, the coupling of the three-layer structure is looser, which is more convenient for modification.

If the number of databases, users and aircraft companies of the system is further increased, the two-tier client will be easier in data management and database sharing due to its simple task. However, due to the heavy task of logical processing, the page loading will be further slow, which will make the user experience decline.

6. Program Design Description

6.1 Database connection

Connect to my local database and enter my account password (create java Dao class + servlet)



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
homework_13_at_night src net javaguides login database LoginDao
Project homework_13_at_night D:\semest...
  .idea
  out
  src
    netjavaguides.login
      bean
        LoginBean
      database
        LoginDao
      web
        LoginServlet
    web
      inc
      WEB-INF
        index.jsp
        login.jsp
        loginsuccess.jsp
        homework_13_at_night.html
  External Libraries
  Scratches and Consoles
  Favorites
  Web

1 package net.javaguides.login.database;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8
9 import net.javaguides.login.bean.LoginBean;
10
11 public class LoginDao {
12
13     public boolean validate(LoginBean loginBean) throws ClassNotFoundException {
14         boolean status = false;
15
16         Class.forName("com.mysql.jdbc.Driver");
17
18         try (Connection connection = DriverManager
19             .getConnection("jdbc:mysql://localhost:3306/airline_databases?useSSL=false", "root", "root1234")) {
20
21             // Step 2:Create a statement using connection object
22             PreparedStatement preparedStatement = connection
23                 .prepareStatement("select * from login where username = ? and password = ? ");
24             preparedStatement.setString(1, loginBean.getUsername());
25             preparedStatement.setString(2, loginBean.getPassword());
26
27             System.out.println(preparedStatement);
28             ResultSet rs = preparedStatement.executeQuery();
29             status = rs.next();
30
31         } catch (SQLException e) {
32             // process sql exception
33             printSQLException(e);
34         }
35     }
36
37     return status;
38 }
```

6.2 System menu

1. Open the reservation form-

- Select the flight
- Destination
- Ticket fair
- Seat
- Passenger details etc.

2. System program-

- Query of airline has its own database.
- When merging flights, the time required for connecting airports

should be considered, i.e. connection time. When booking all flights of the same airline, the price will be discounted according to the airline's own discount (%).

AIRLINE RESERVATION SYSTEM

RESERVATION FORM

RESERVATION DETAILS

FLIGHT NO :	<input type="text"/>	TICKET FARE :	<input type="text"/>
ORIGIN :	<input type="text"/>	FLIGHT TYPE :	<input type="text"/>
DESTINATION :	<input type="text"/>		
CLASS :	<input type="text"/>		

PASSENGER DETAILS

PASSENGER NAME :	<input type="text"/>	<input type="button" value="Adodc2"/> <input type="button" value="Adodc2"/>
PASSENGER ADDRESS :	<input type="text"/>	
PASSPORT NO :	<input type="text"/>	
PASSENGER STATUS :	<input type="text"/>	

6.3 User login

First we have to login using the username and password. Then we can see the reservation form.

USER NAME :	<input type="text"/>
PASSWORD :	<input type="text"/>

6.4 Data entry, modification and deletion

Data Entry: Here we have to enter all the data according to our choice and then click the “ADD” button. As a result, all the data will be entered in the database.

AIRLINE RESERVATION SYSTEM
RESERVATION FORM

RESERVATION DETAILS

FLIGHT NO :	B-757	TICKET FARE :	150USD
ORIGIN :	WUHAN	FLIGHT TYPE :	FFES
DESTINATION :	BEIJING		
CLASS :	ECONOMY-CLASS		

PASSENGER DETAILS

PASSENGER NAME :	ZAMAN	<input type="button" value="Adodc2"/>
PASSENGER ADDRESS :	HONGSHAN, WUHAN	
PASSPORT NO :	E-32223	
PASSENGER STATUS :	MALE	

CLICK ADD TO INSERT DATA

AIRLINE RESERVATION SYSTEM	
RESERVATION FORM	
RESERVATION DETAILS	
FLIGHT NO : B-757	TICKET FARE : 150USD
ORIGIN : WUHAN	FLIGHT TYPE : FFES
DESTINATION : BEIJING	
CLASS : ECONOMY-CLASS	
PASSENGER DETAILS	
PASSENGER NAME : ZAMAN	
PASSENGER ADDRESS : HONGSHAN, WUHAN	
PASSPORT NO : E-32223	
PASSENGER STATUS : MALE	
EDIT	CANCEL

Modification: For data modification click the “EDIT” button. Then we modified some data and click the “ADD” button.

AIRLINE RESERVATION SYSTEM	
RESERVATION FORM	
RESERVATION DETAILS	
FLIGHT NO : B-757	TICKET FARE : 150USD
ORIGIN : WUHAN	FLIGHT TYPE : FFES
DESTINATION : BEIJING	
CLASS : ECONOMY-CLASS	
PASSENGER DETAILS	
PASSENGER NAME : ZAMAN	
PASSENGER ADDRESS : HONGSHAN, WUHAN	
PASSPORT NO : E-32223	
PASSENGER STATUS : MALE	
EDIT	CANCEL
CLICK EDIT TO MODIFICATION DATA	

AIRLINE RESERVATION SYSTEM
RESERVATION FORM

RESERVATION DETAILS

FLIGHT NO :	B-757	TICKET FARE :	250USD
ORIGIN :	WUHAN	FLIGHT TYPE :	FFES
DESTINATION :	BEIJING		
CLASS :	BUSINESS-CLASS		

PASSENGER DETAILS

PASSENGER NAME :	ZAMAN	Adodc2
PASSENGER ADDRESS :	HONGSHAN, WUHAN	◀ ▶
PASSPORT NO :	E-32223	
PASSENGER STATUS :	MALE	

CLICK ADD TO INSERT DATA

ADD **view** **EXIT**

After Modification:

AIRLINE RESERVATION SYSTEM
RESERVATION FORM

RESERVATION DETAILS

FLIGHT NO : B-757	TICKET FARE : 250USD
ORIGIN : WUHAN	FLIGHT TYPE : FFES
DESTINATION : BEIJING	
CLASS : BUSINESS-CLASS	

PASSENGER DETAILS

PASSENGER NAME : ZAMAN
PASSENGER ADDRESS : HONGSHAN, WUHAN
PASSPORT NO : E-32223
PASSENGER STATUS : MALE

EDIT **CANCEL**

Deletion: For delete all the data we have to click the “CANCEL” button.

As a result, all the data will be deleted from the database.

AIRLINE RESERVATION SYSTEM	
RESERVATION FORM	
RESERVATION DETAILS	
FLIGHT NO : B-757	TICKET FARE : 250USD
ORIGIN : WUHAN	FLIGHT TYPE : FFES
DESTINATION : BEIJING	
CLASS : BUSINESS-CLASS	
PASSENGER DETAILS	
PASSENGER NAME : ZAMAN	
PASSENGER ADDRESS : HONGSHAN, WUHAN	
PASSPORT NO : E-22222	
PASSENGER STATUS : MALE	
EDIT	CANCEL
CLICK CANCEL TO DELETE DATA	

6.5 Database query

User Login: After enter the username and password we can see the user data in the database.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'airline_databases' database is selected. In the center pane, a query window displays the following SQL code:

```
SQLQuery1.sql - D:\UH3fahim (54)
SELECT [username], [password]
FROM [airline_databases].[dbo].[UserAccount]
WHERE username='fahim' and password='fm6fm6';
```

The results pane shows a single row of data:

username	password
fahim	fm6fm6

The status bar at the bottom indicates "Replace all 'useraccount', 'UserAccount', 'Hidden, Current Document".

Data Entry: After enter all the data in the reservation form we can see all the data in the database.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'airline_databases' database is selected. In the center pane, a query window displays the following SQL code:

```
SQLQuery1.sql - D:\UH3fahim (52)
SELECT [Name], [Address], [PassportNo], [Status], [FlightNo], [FlightType], [Origin], [Destination], [Seat], [TicketFee]
FROM [airline_databases].[dbo].[Reservation]
WHERE [Name] = 'ZAMAN' and [Address] = 'HONGSHAN, WUHAN' and [PassportNo] = 'E-32223' and [Status] = 'MALE' and [FlightNo] = 'B-757' and [FlightType] = 'FFS'
```

The results pane shows a single row of data:

Name	Address	PassportNo	Status	FlightNo	FlightType	Origin	Destination	Seat	TicketFee
ZAMAN	HONGSHAN, WUHAN	E-32223	MALE	B-757	FFS	WUHAN	BEIJING	ECONOMY-CLASS	150USD

The status bar at the bottom indicates "Query executed successfully."

Modification: After modify some data in the reservation form we can see

the modified data in the database.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'airline_databases' database is selected. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3HVahim (52)' contains the following SQL code:

```
SELECT [Name], [Address], [PassportNo], [Status], [FlightNo], [FlightType], [Origin], [Destination], [Seat], [TicketFare]
FROM [airline_databases].[dbo].[Reservation]
WHERE Name='ZAMAN' and Address='HONGSHAN, WUHAN' and PassportNo='E-32223' and Status='MALE' and FlightNo='B-757' and FlightType='FFS'
```

The results pane shows a single row of data:

Name	Address	PassportNo	Status	FlightNo	FlightType	Origin	Destination	Seat	TicketFare
ZAMAN	HONGSHAN, WUHAN	E-32223	MALE	B-757	FFS	WUHAN	BEIJING	BUSINESS-CLASS	250USD

At the bottom of the interface, a message bar indicates: 'Query executed successfully.' and 'Ln 4 Col 1 Ch 1 INS'.

Deletion: After delete all the data in the reservation form we can't see the

data in the database anymore.

The screenshot shows the Microsoft SQL Server Management Studio interface. In the Object Explorer, the 'airline_databases' database is selected. In the center pane, a query window titled 'SQLQuery1.sql - D...OH3HVahim (52)' contains the same SQL code as the previous screenshot:

```
SELECT [Name], [Address], [PassportNo], [Status], [FlightNo], [FlightType], [Origin], [Destination], [Seat], [TicketFare]
FROM [airline_databases].[dbo].[Reservation]
WHERE Name='ZAMAN' and Address='HONGSHAN, WUHAN' and PassportNo='E-32223' and Status='MALE' and FlightNo='B-757' and FlightType='FFS'
```

The results pane is empty, indicating '0 rows'.

At the bottom of the interface, a message bar indicates: 'Query executed successfully.' and 'Ln 3 Col 224 Ch 224 INS'.

7. Summary of Course Design

7.1 Problems and solutions in curriculum design

1. Haven't deep knowledge about store process in database. After complete this project, know me able to connect, modify the remote connection.
2. Now I am able to insertion and values. And have knowledge modify values which are already inserted in a table.
3. If has there any kind of type defined error like string, varchar, char, int, float etc. Now I am able to resolve.

7.2 Analysis of existing problems

In this database project (airplane database management) we are used to insertion data in our local database directly (means an operators only they can insert).

But I future, it's can update like real time data insertion.

If we create an application it's can be able to call (.json) API from Airlines Company and airport management center. And then our application can modify API data and convert it SQL language and real-time dynamic insert, modify and deletion on database.

7.3 Experience of curriculum design

In the project curriculum design

- Connect a local database software like Ms SQL 2008, MySQL etc.
- Create/build database
- Create table
- Create schema
- Create or make relationship between a table to another table
- Insert data, modification and deletion by (database management operator)
- Passenger query

I also learn java (database connection servlet and Dao class) how to create a java class and how connect a. In additional to these, how to overcome my database by Dao class laziness, How to find a quick and accurate solution using the Internet (Google). Special from <https://stackoverflow.com/> how to get my answer. And a best website for learning Database is <https://www.geeksforgeeks.org/dbms/>