



湖北工业大学
HUBEI UNIVERSITY OF TECHNOLOGY

DBMS

Course Design Report

Design Topic: online airplane management database

Mentor Assistant teacher of prof. Chen jianxia

Profession prof. Chen jianxia

Class 19lc software engineering

Student ID 1811562127

Name ZUBAYER S M(苏贝尔)

Date 14-12-2020

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

System operation environment description

System overview

*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 (%).*

System requirements analysis Requirement

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

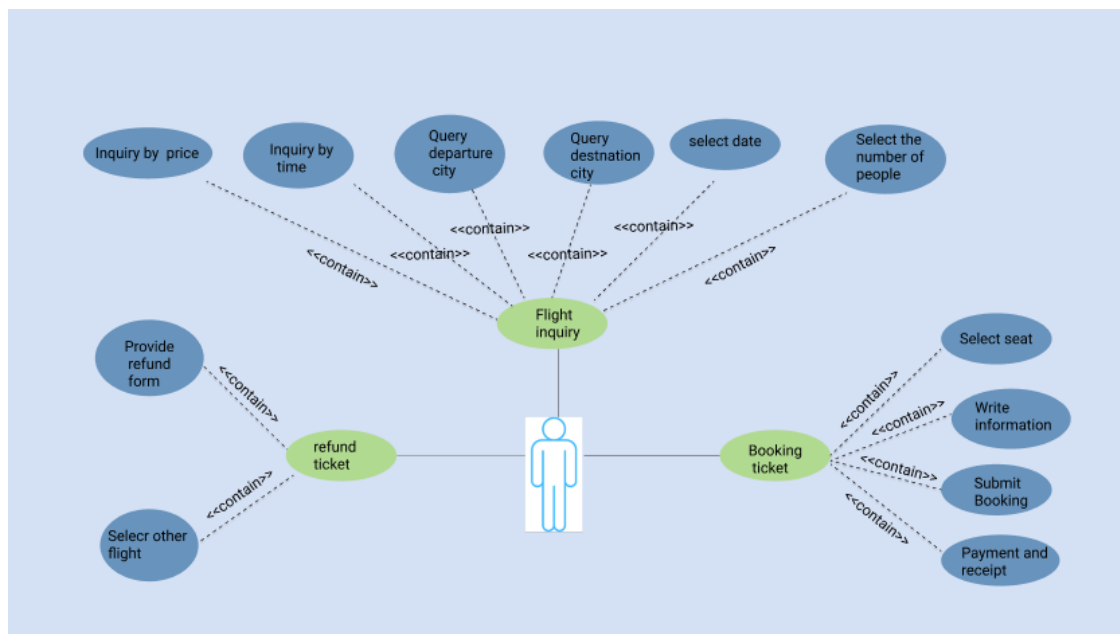
1.2 team division

This project made by 苏贝尔. 1811562127

Organization chart and Data flow graph and data dictionary

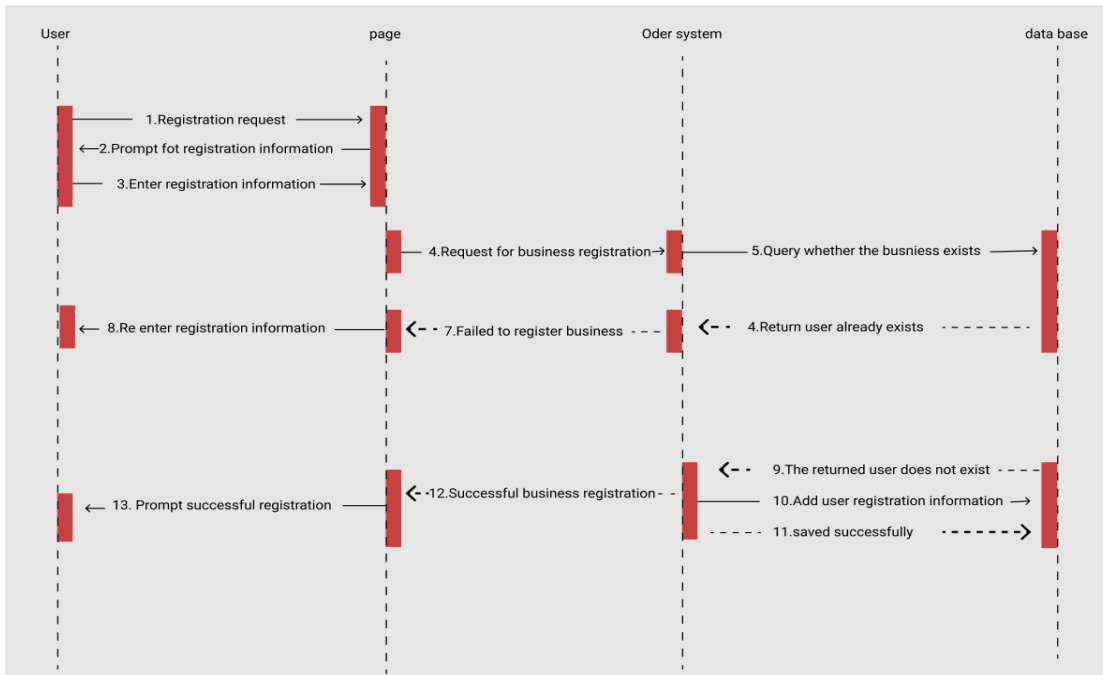
3 UML modeling

3.1 Use case diagram

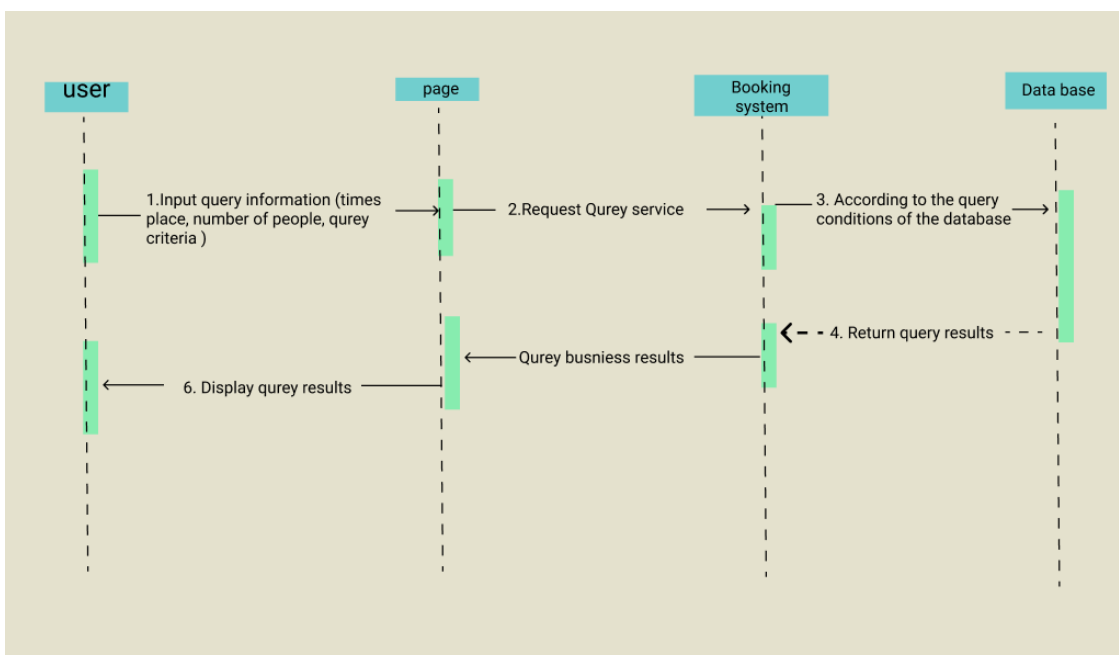


3.2 Three layer C / S structure

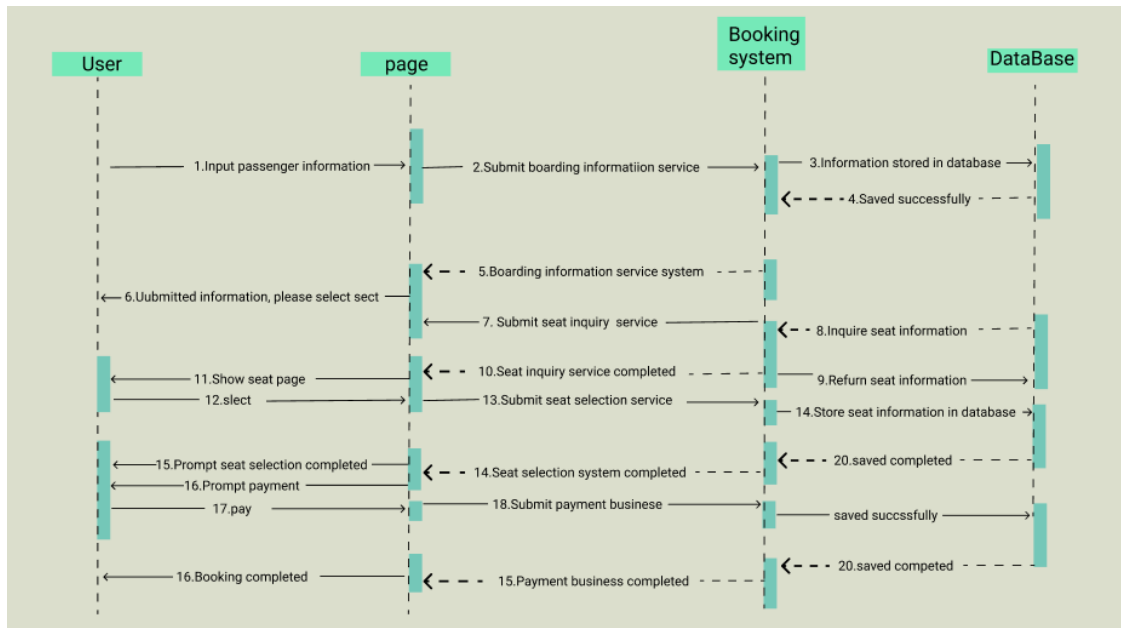
3.2.1 user registration sequence diagram



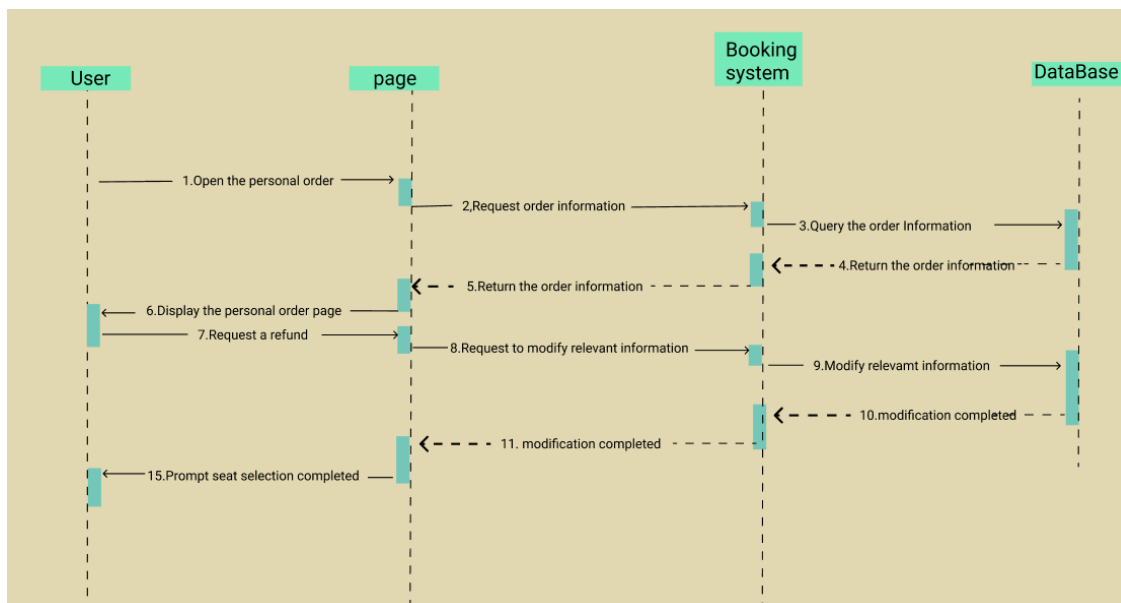
3.2.2 flight query sequence diagram



3.2.3 User booking sequence diagram

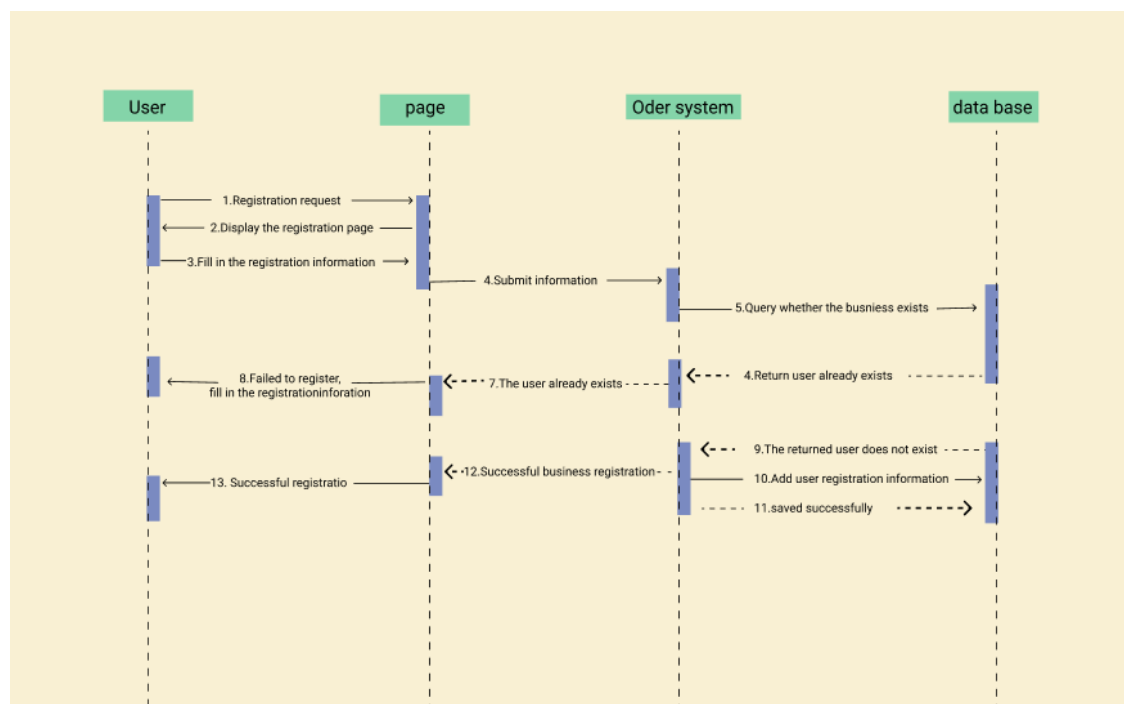


3.2.4 User refund sequence diagram

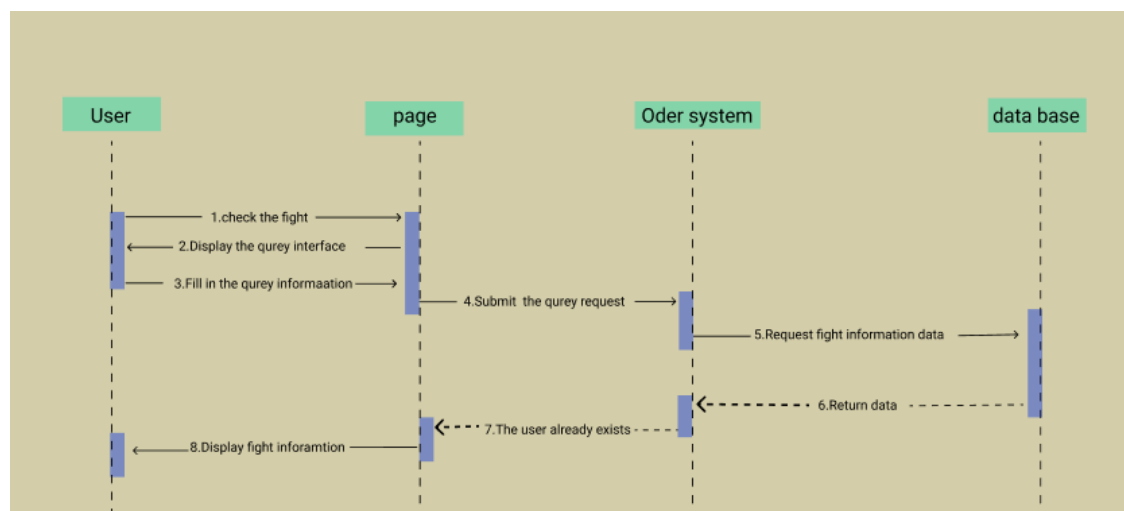


3.3 two-layer C / S structure

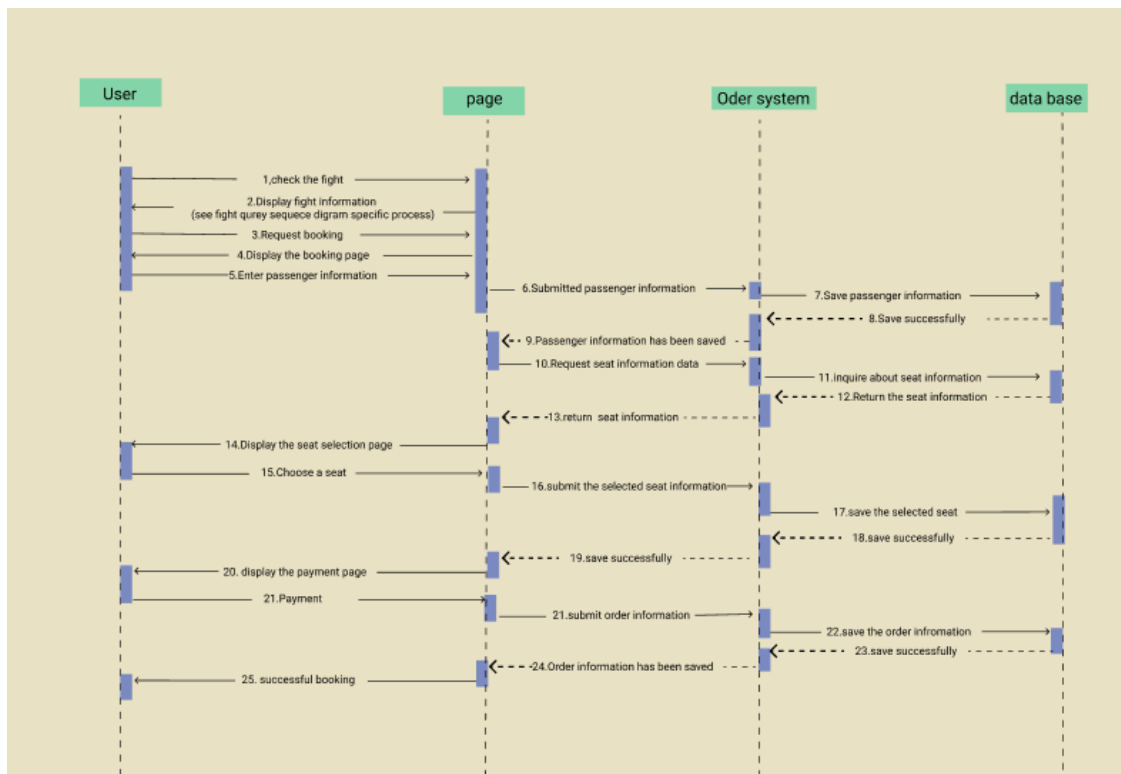
3.3.1 User registration timing chart



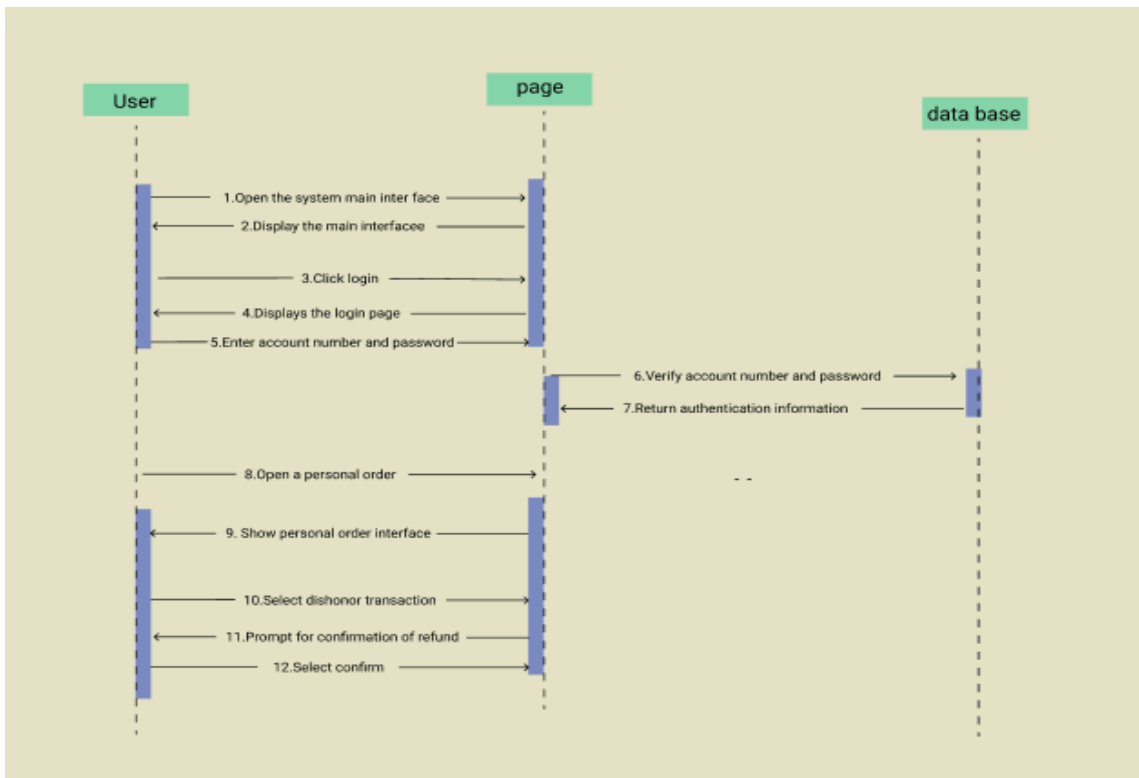
3.3.2 Fight inquiry sequence diagram



3.3.3 User booking sequence diagram



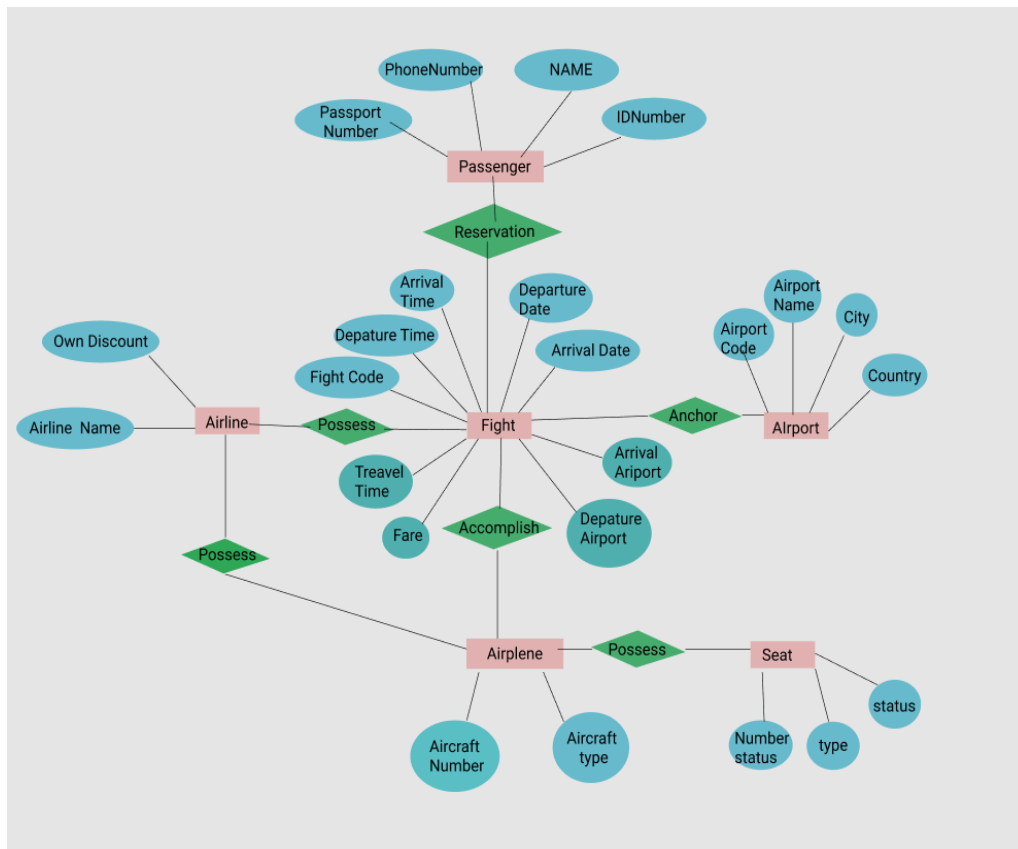
3.3.4 User refund sequence diagram



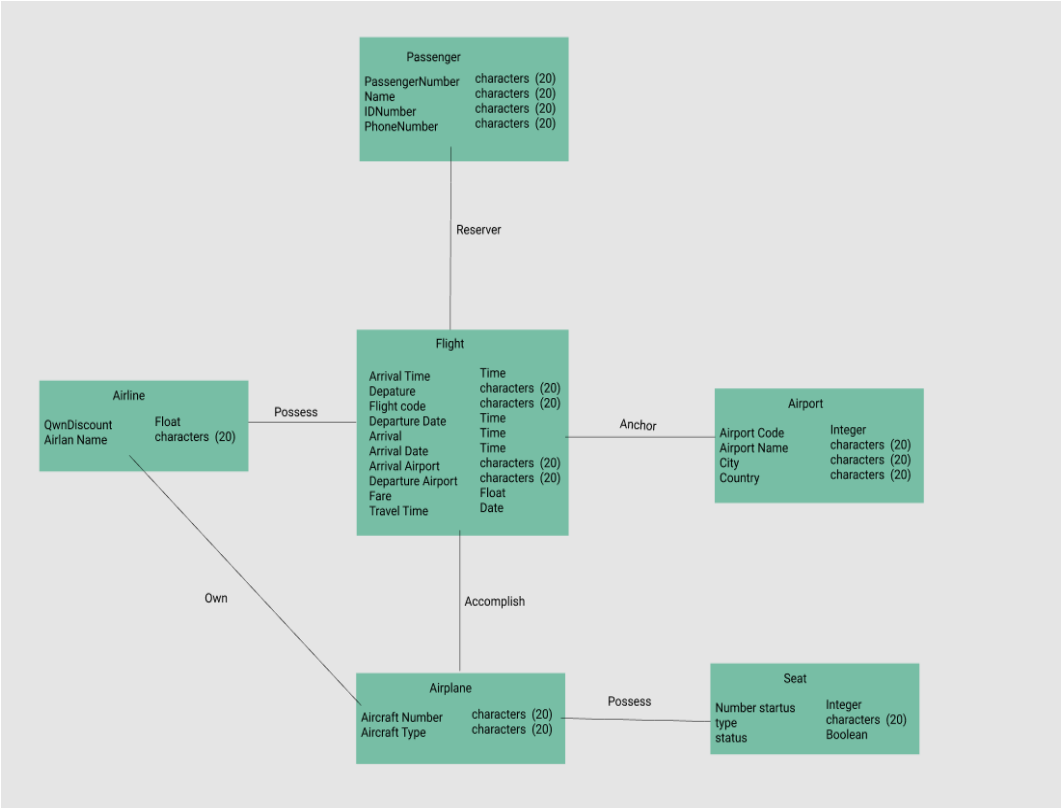
Database design

4.1 Conceptual structure design

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)

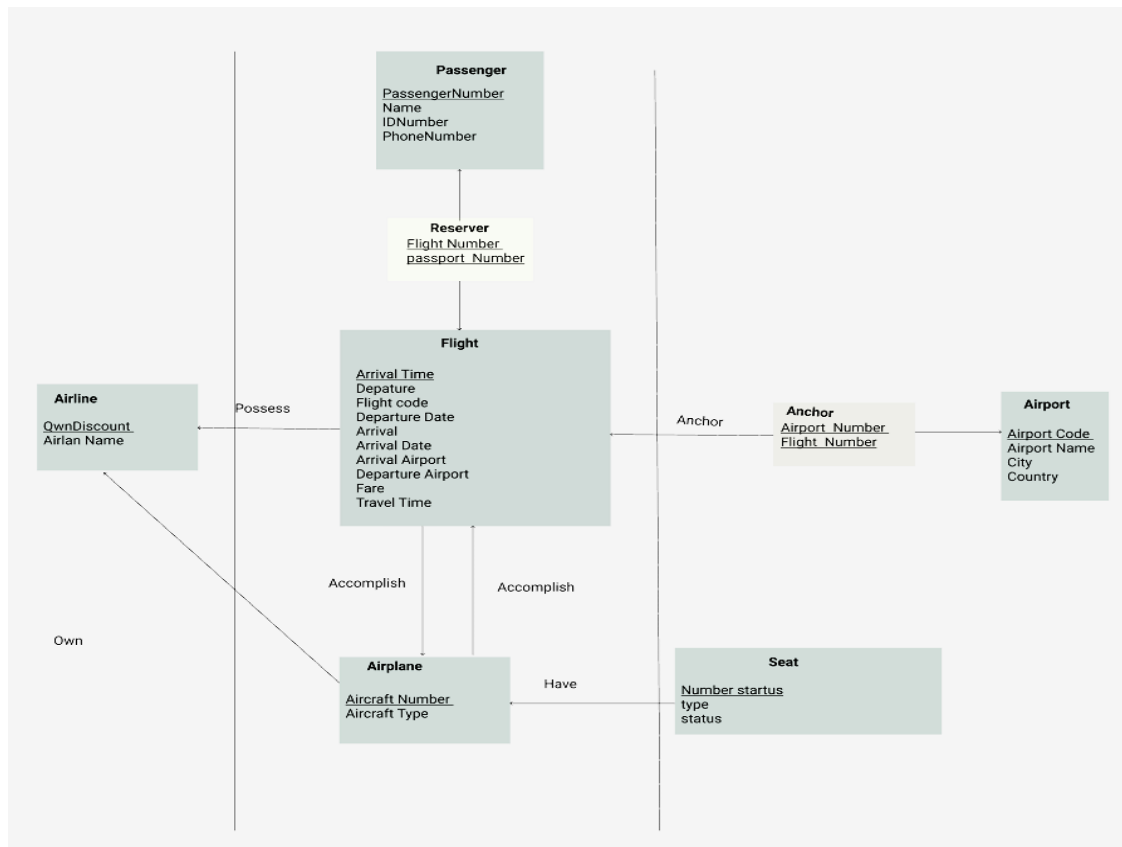
Airport (airport number, airport name, city, country)

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

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

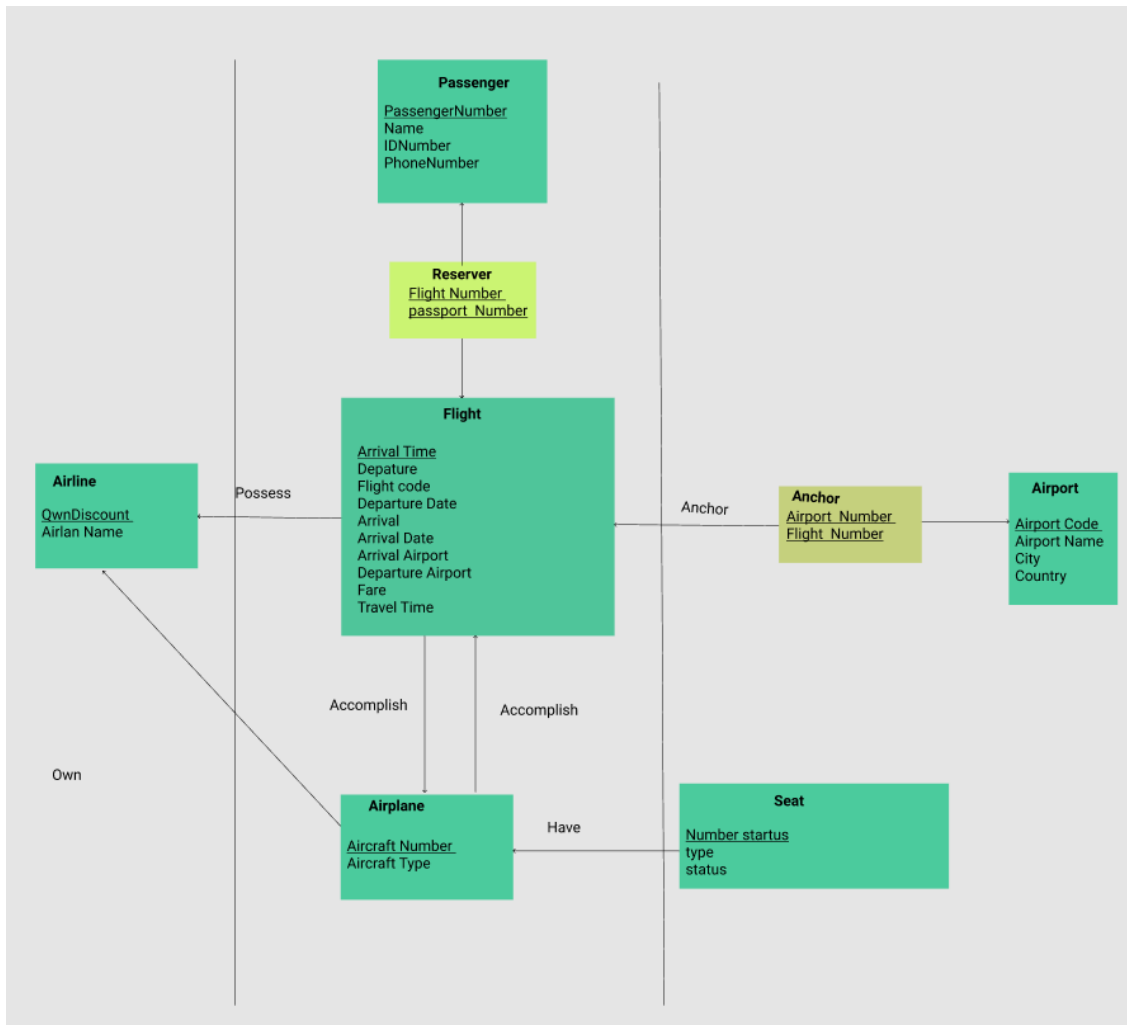
In the above relational mode, the underline is the main code of horizontal line, and the underline is the outer code of wavy line

Logical structure diagram of design:



4.3 Physical structure design

Physical structure diagram of design:



5. Database implementation and operation

5.1 System design and Implementation

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

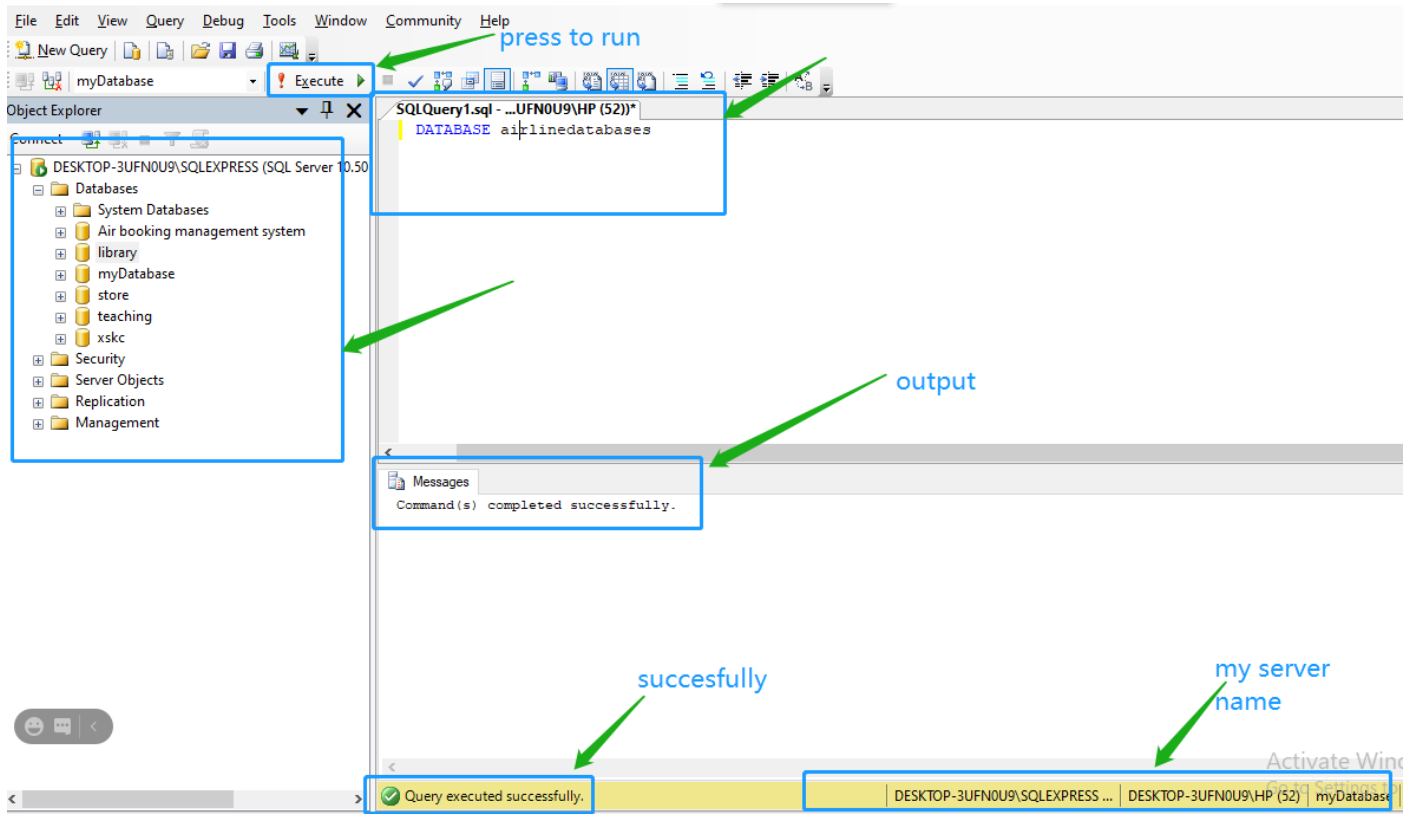
front-end page is designed, and then according to the data required by the front-end page, the servlet is used to process the data retrieved from the database and the data returned from the front-end page to realize the interaction between the front-end and the front-end.

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:



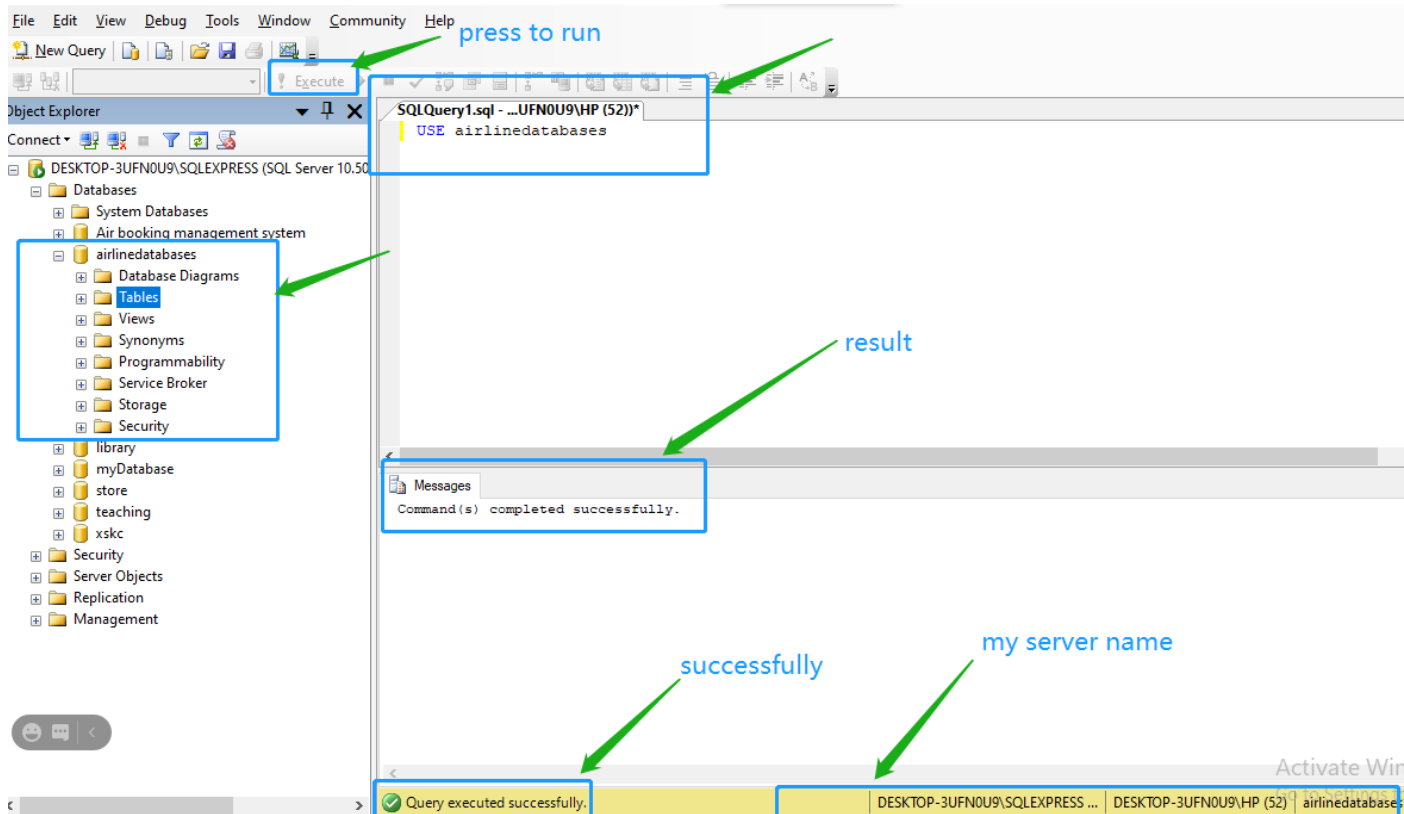
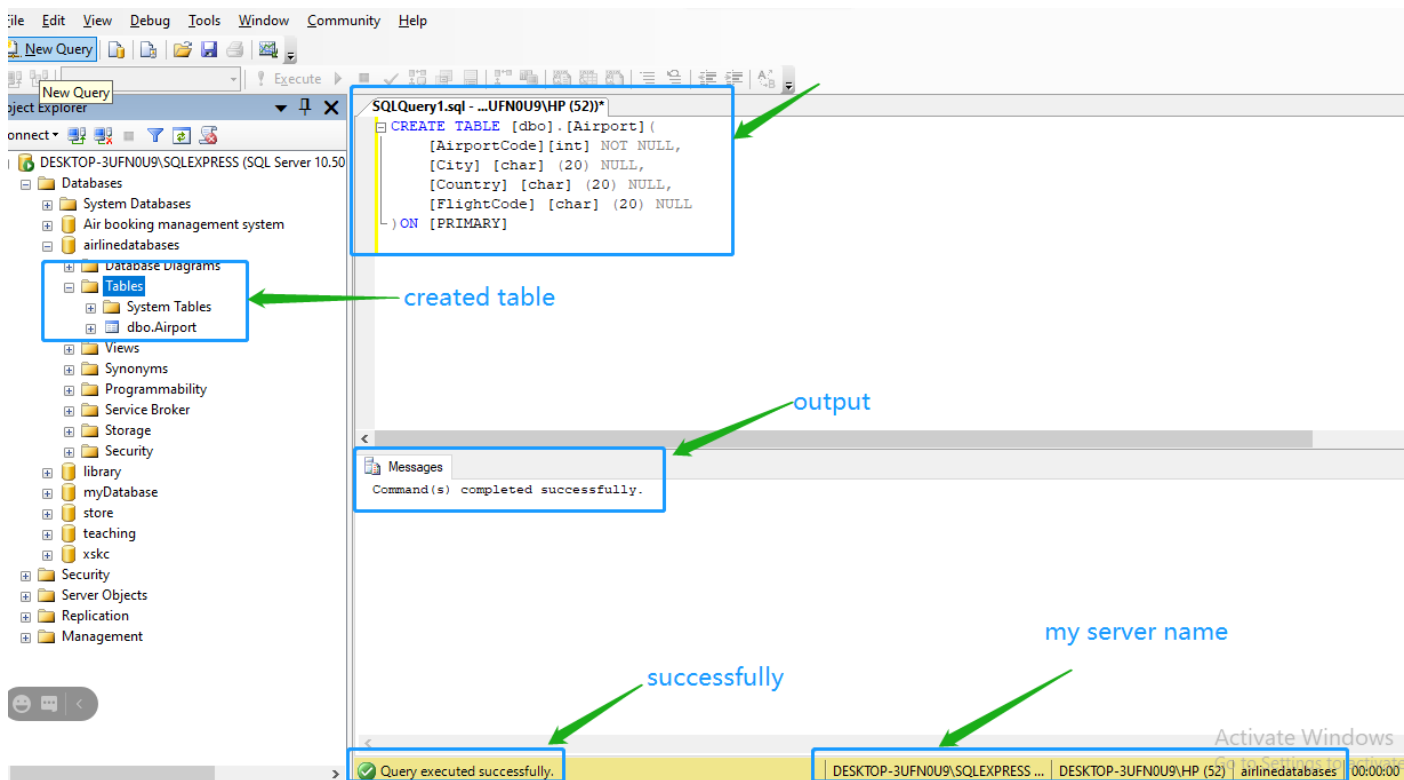
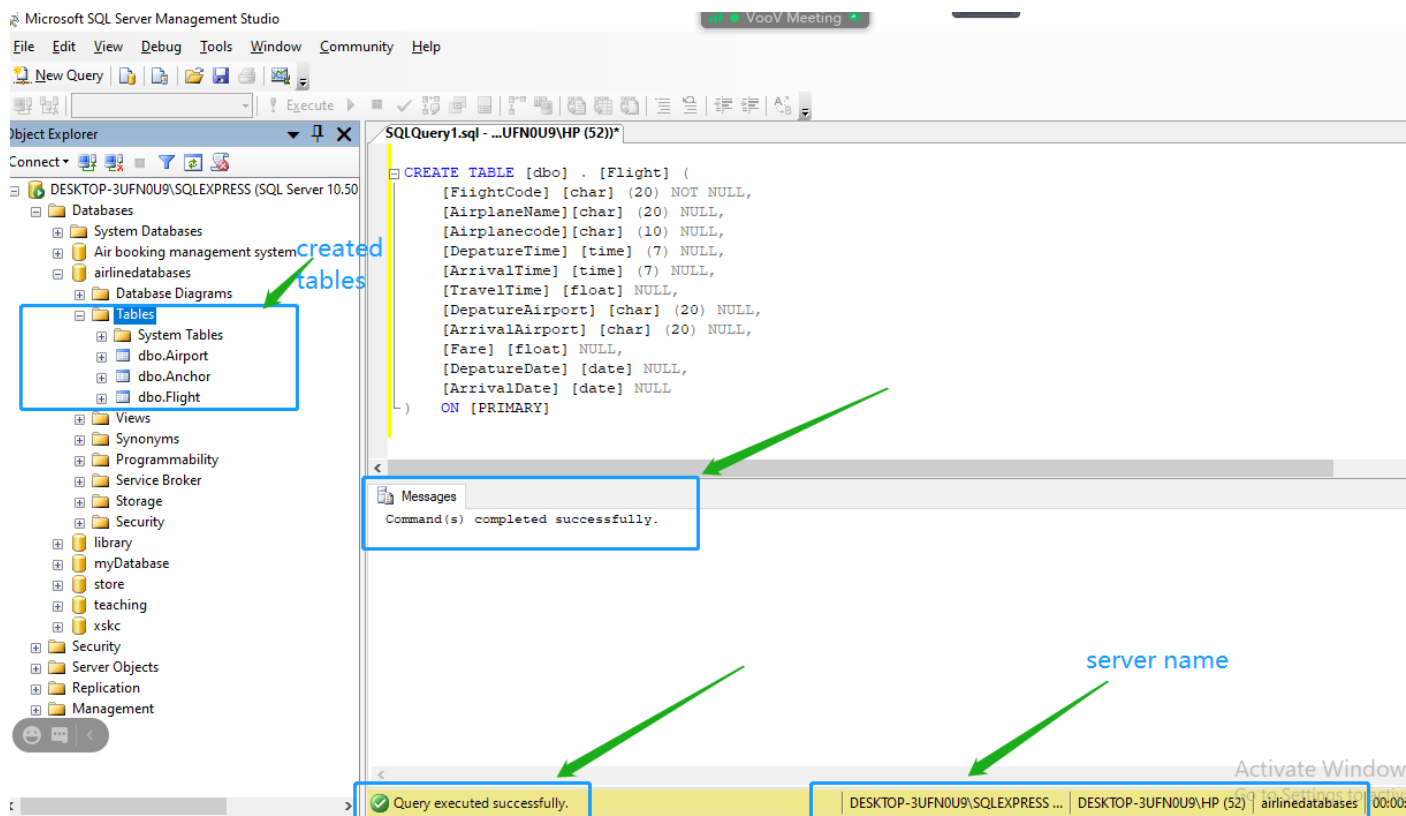
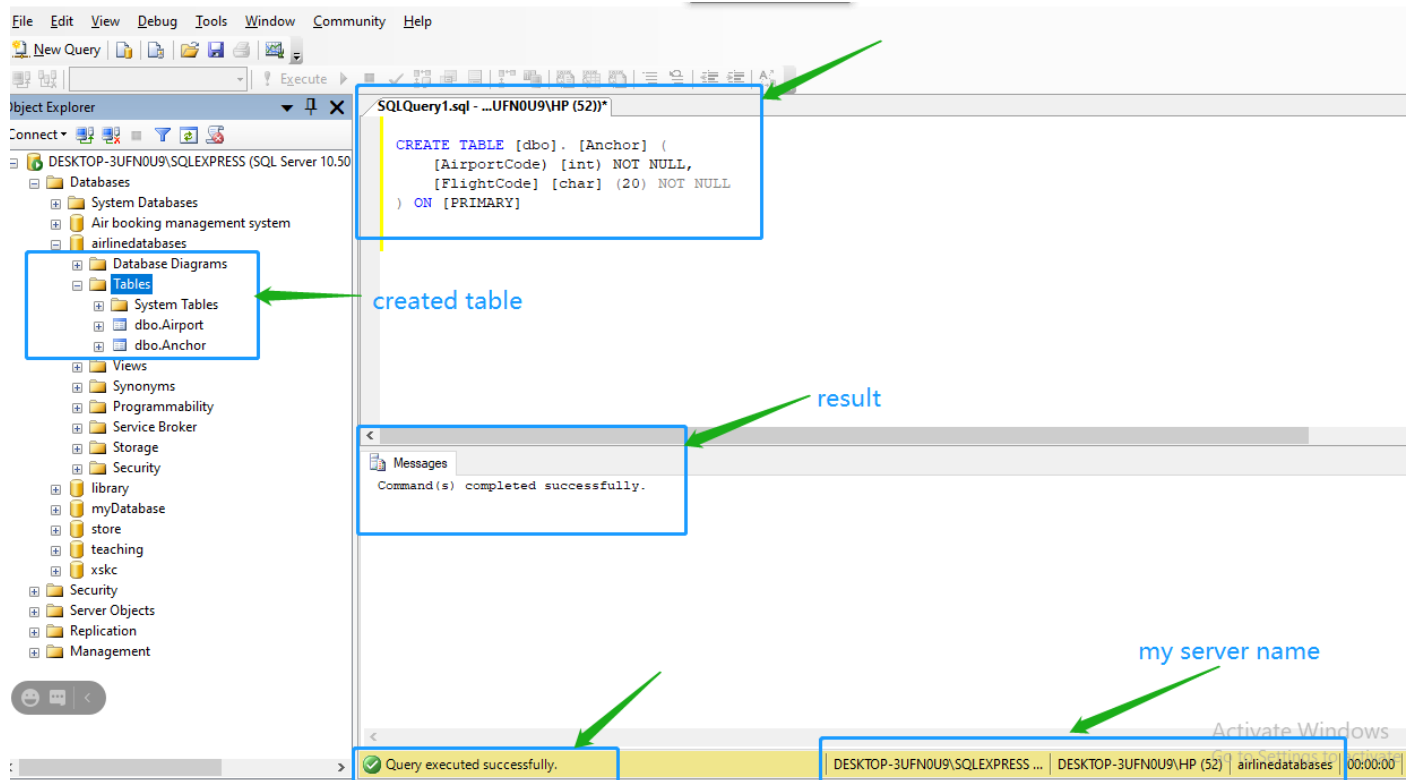
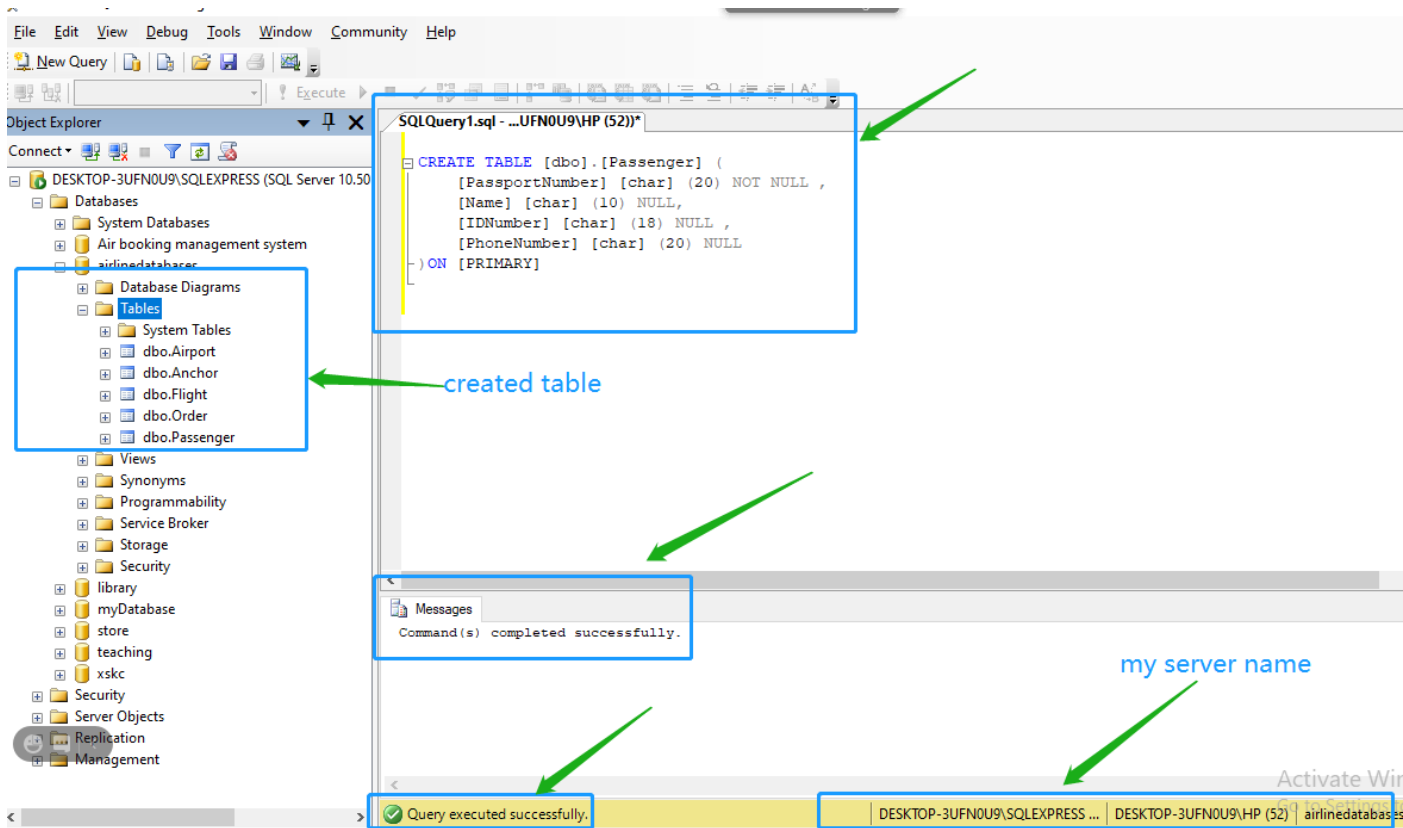
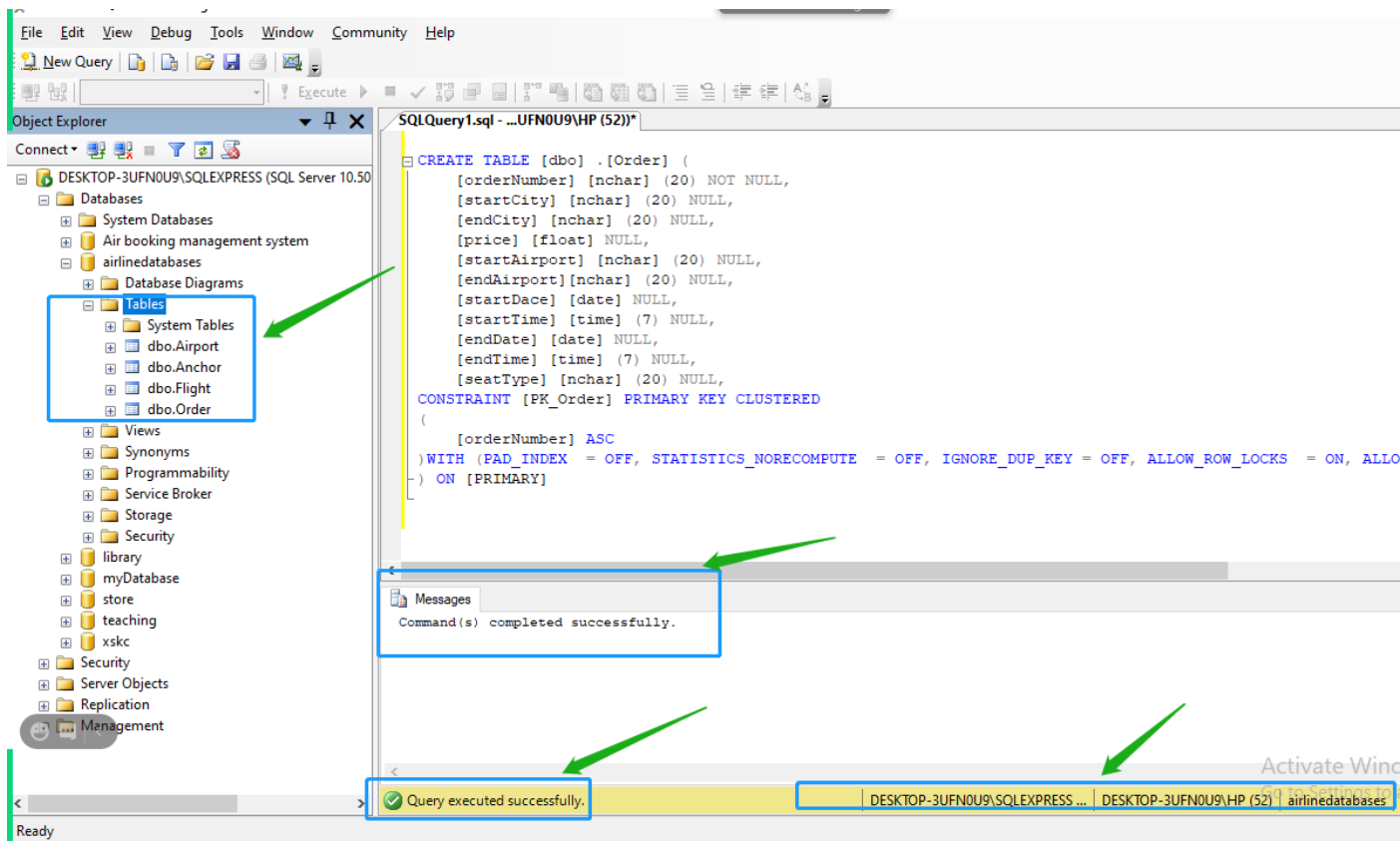


Table creation:







File Edit View Debug Tools Window Community Help

New Query

press to run

Execute

Object Explorer

Connect

DESKTOP-3UFN0U9\SQLEXPRESS (SQL Server 10.0.5008.8)

Databases

- System Databases
- Air booking management system
- airlinedatabases
 - Database Diagrams
 - Tables
 - dbo.Airline
 - dbo.Airplane
 - dbo.Airport
 - dbo.Anchor
 - dbo.Flight
 - dbo.Order
 - dbo.Passenger
 - dbo.Seat
 - dbo.useraccount
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- library
- myDatabase
- store
- teaching
- xskc
- Security
- Server Objects

SQLQuery1.sql - ...UFN0U9\HP (52)*

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

created tables

result

Messages

Command(s) completed successfully.

my server name

Query executed successfully.

DESKTOP-3UFN0U9\SQLEXPRESS ... | DESKTOP-3UFN0U9\HP (52) | airlinedatabases

File Edit View Debug Tools Window Community Help

New Query

Execute

Object Explorer

Connect

DESKTOP-3UFN0U9\SQLEXPRESS (SQL Server 10.0.5008.8)

Databases

- System Databases
- Air booking management system
- airlinedatabases
 - Database Diagrams
 - Tables
 - dbo.Airline
 - dbo.Airplane
 - dbo.Airport
 - dbo.Anchor
 - dbo.Flight
 - dbo.Order
 - dbo.Passenger
 - Views
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
- library
- myDatabase
- store
- teaching
- xskc
- Security
- Server Objects
- Replication
- Management

SQLQuery1.sql - ...UFN0U9\HP (52)*

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

created tables

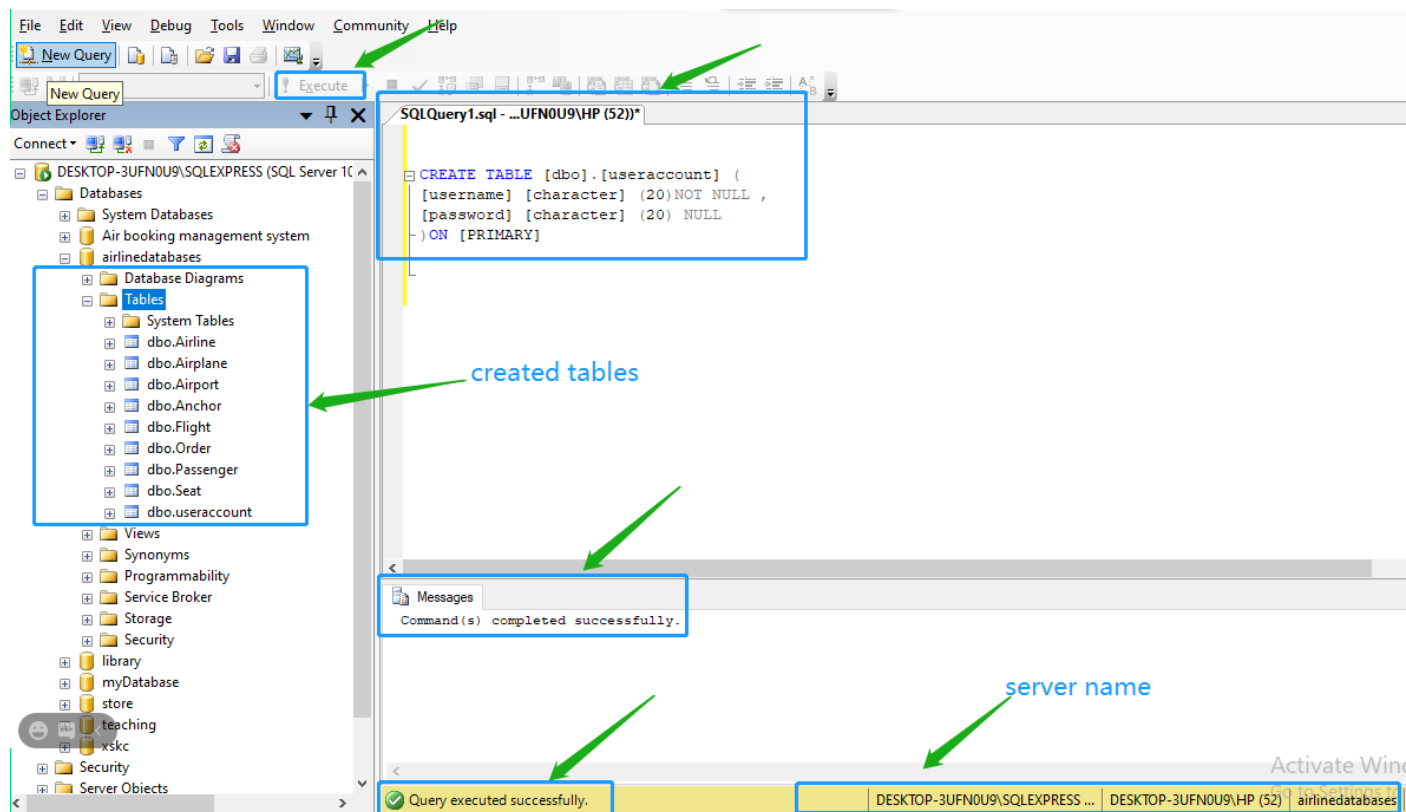
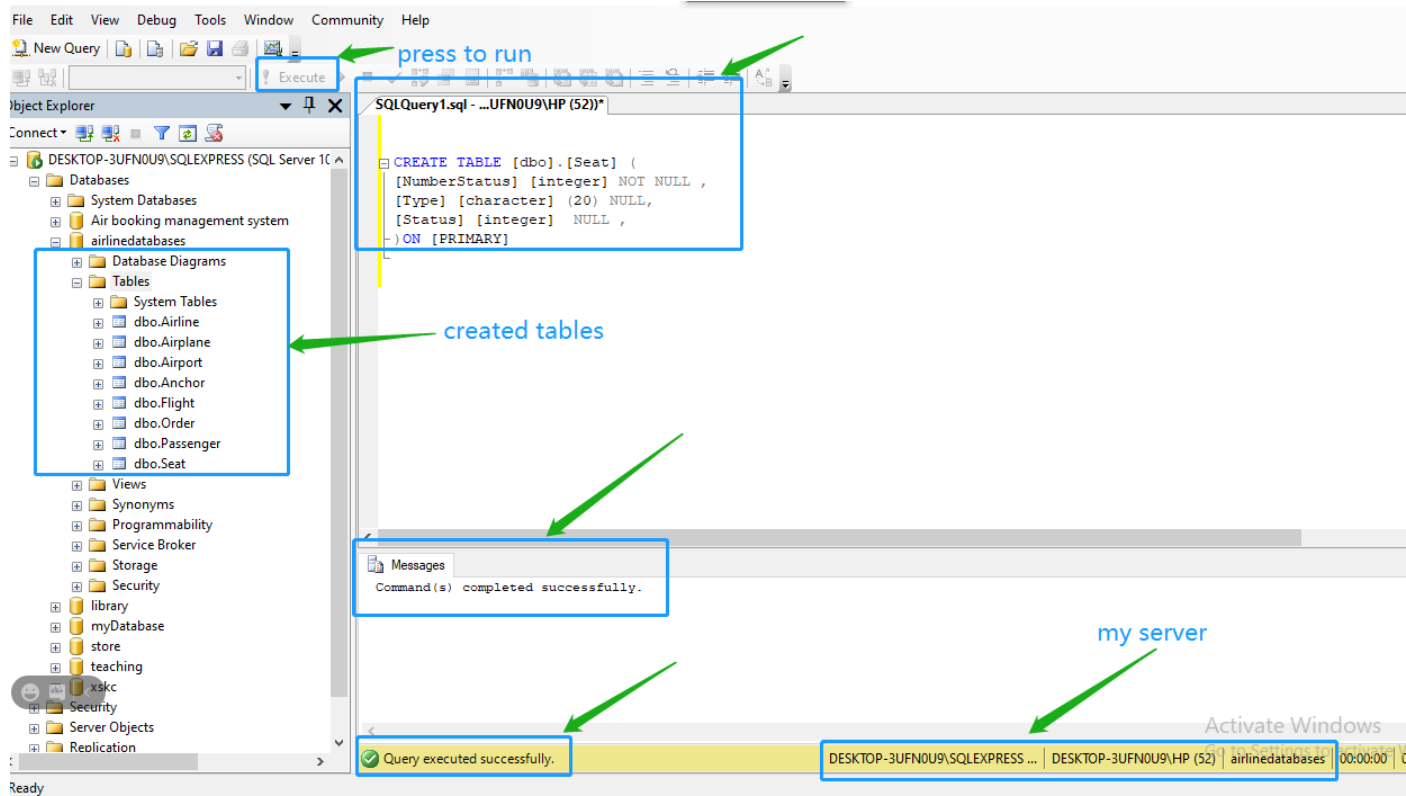
Messages

Command(s) completed successfully.

server name

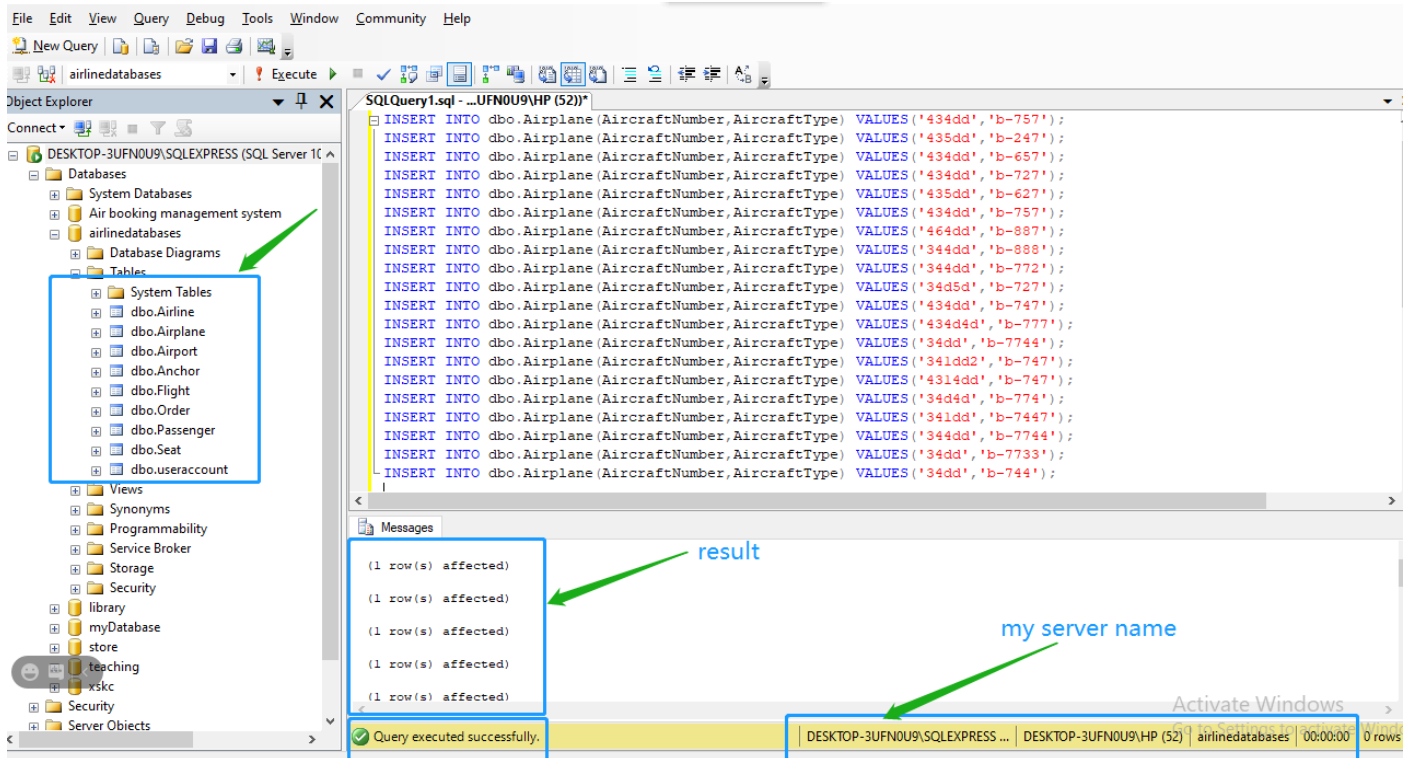
Query executed successfully.

DESKTOP-3UFN0U9\SQLEXPRESS ... | DESKTOP-3UFN0U9\HP (52) | airlinedatabases

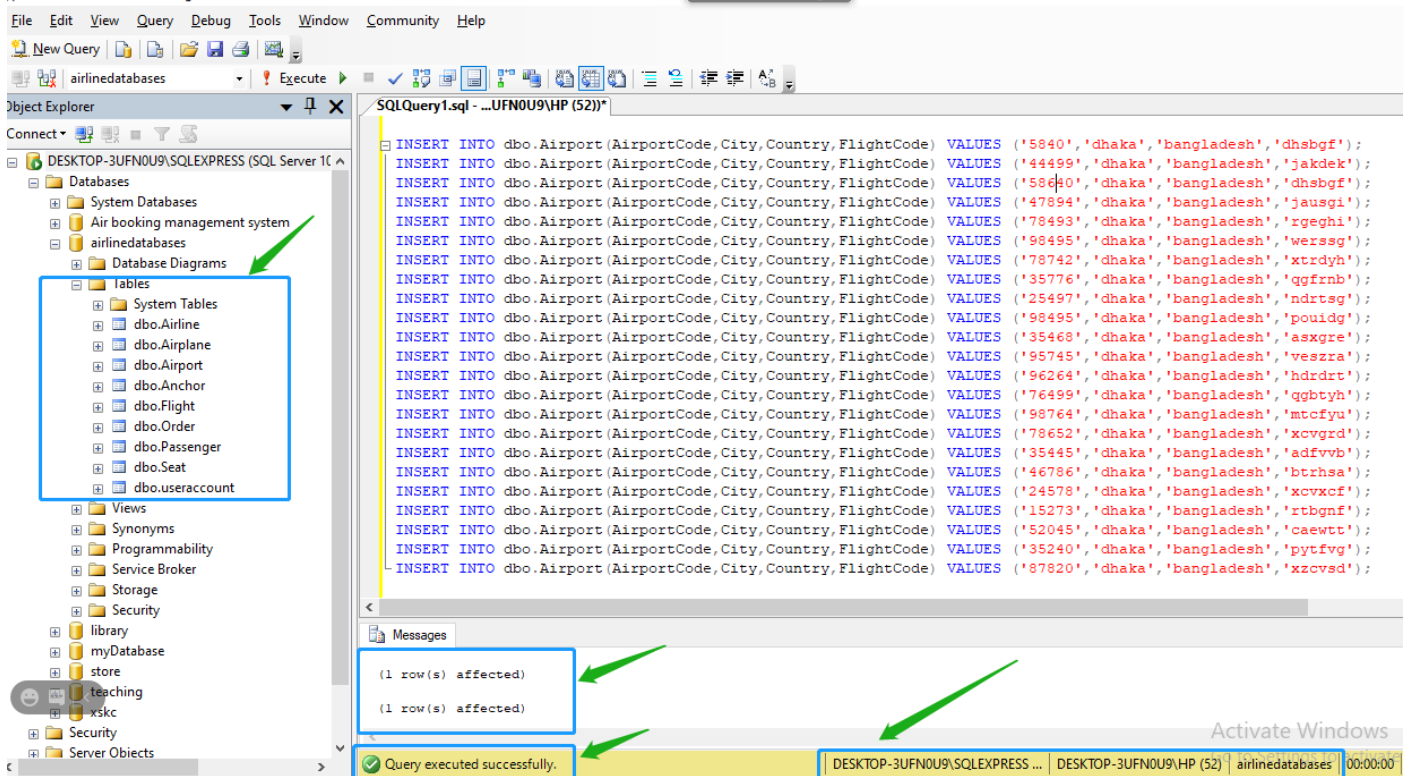


##(part -02) Database query:

Data insert at airplane table



Data insert at airport table



Data insert at Flight table

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'airlinedatabases'. The 'Tables' folder is expanded, and the 'dbo.Flight' table is selected. The main window shows the SQL query editor with the following INSERT statements:

```
INSERT INTO dbo.Flight (FlightCode, AirplaneName, AirplaneCode, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
VALUES ('FFES', 'China southern', '44tdw', '10:34:09 AM', '7:04:09 AM', '3.2', 'Beijing', 'Wuhan', '342.3', '1-28-2021', '1-28-2021');

INSERT INTO dbo.Flight (FlightCode, AirplaneName, AirplaneCode, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
VALUES ('FFES', 'Eastern Airlines', '46tgf', '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, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
VALUES ('KLHL', 'China southern', '44tdw', '10:34:09 PM', '7:04:09 AM', '1.5', 'Wuhan', 'Beijing', '342.3', '6-2-2021', '6-2-2021');

INSERT INTO dbo.Flight (FlightCode, AirplaneName, AirplaneCode, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
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, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
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, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
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, DepartureTime, ArrivalTime, TravelTime, DepartureAirport, ArrivalAirport)
VALUES ('SGGV', 'China southern', '479DER', '10:34:09 AM', '3:02:09 PM', '3.2', 'Wuhan', 'Pakistan', '342.3', '1-28-2021', '1-28-2021');
```

The Messages window shows the results of the query execution:

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

The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-3UFN0U9\SQLEXPRESS ...'.

Data insert for Passenger table

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'airlinedatabases'. The 'Tables' folder is expanded, and the 'dbo.Passenger' table is selected. The main window shows the SQL query editor with the following INSERT statements:

```
INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('E-3223', 'ZAMAN', '493882NJD', '546');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('T-4245', 'MANIK', '493882NJD', '5658');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('S45W62', 'TOMOY', '493882NJD', '5764');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('C-4567', 'MAMUN', '493882NJD', '9687');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('H-553G', 'PIAS', '493882NJD', '36756');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('CE5B35', 'ZUBAYER', '493882NJD', '38');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('E-53B55', 'SARKAR', '493882NJD', '86');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('AD56B53', 'KHAN', '493882NJD', '6756');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('E-56563', 'ZAHIN', '493882NJD', '875');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('T-5645', 'BRTAR', '493882NJD', '6786');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('EAD6G3G', 'MADHARCHUD', '493882NJD');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('ET-87785', 'MAGIBETI', '493882NJD');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('CIN-NM3', 'SCHODI', '493882NJD', '86');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('AS-5356B', 'IEKSSS', '493882NJD', '7');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('INE-BVB', 'EMON', '493882NJD', '8675');

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('EA-3355', 'NAIKHAN', '493882NJD', '8);

INSERT INTO dbo.Passenger (PassportNumber, Name, IDNumber, PhoneNumber)
VALUES ('EE-3VBN', 'AMIN KHAN', '493882NJD');
```

The Messages window shows the results of the query execution:

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

The status bar at the bottom indicates 'Query executed successfully.' and 'DESKTOP-3UFN0U9\SQLEXPRESS ...'. A green arrow points to the 'result' text in the Messages window, and another green arrow points to the 'my server name' text in the status bar.

Data insert for Seat table

The screenshot displays the SQL Server Enterprise Manager interface. In the Object Explorer on the left, the 'Tables' folder under 'airlinedatabases' is expanded, and the 'dbo.Seat' table is highlighted with a green arrow. The main window shows a SQL query with 14 INSERT statements for the 'dbo.Seat' table, each inserting a new row with specific values for 'NumberStatus', 'Type', 'Status', and 'ECONOMY-CLASS'. Below the query, the Messages window shows 14 messages, each stating '(1 row(s) affected)', with a green arrow pointing to the list. The status bar at the bottom indicates 'Query executed successfully.' with a green checkmark icon. The taskbar at the bottom shows the active window as 'DESKTOP-3UFN0U9\SQLEXPRESS ...'.

```
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');
```

Messages

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

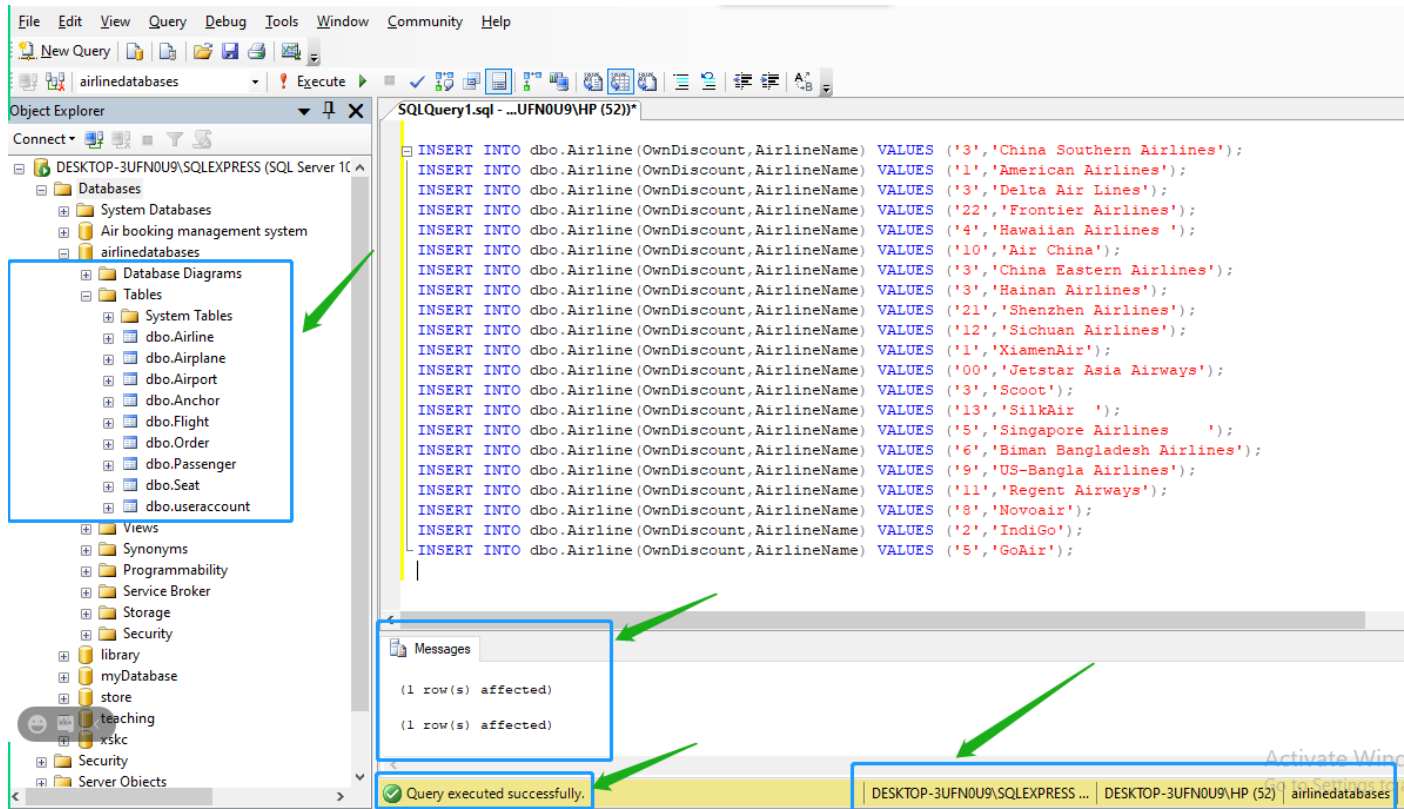
(1 row(s) affected)

(1 row(s) affected)

Query executed successfully.

DESKTOP-3UFN0U9\SQLEXPRESS ... | DESKTOP-3UFN0U9\HP (52) | airlinedatabases | 00:00:00

Data insert for Airline table



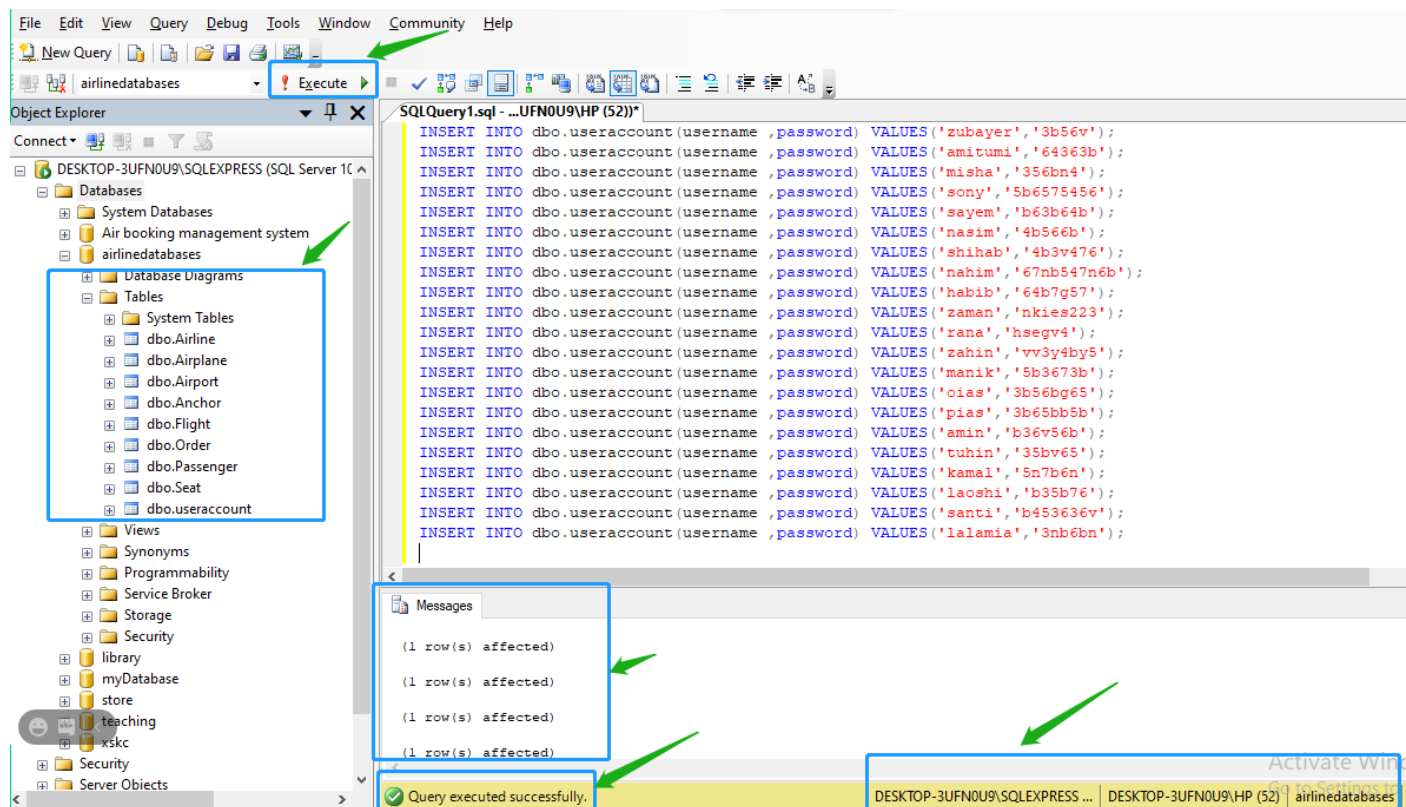
In here I face some problem, its “call string or binary would be truncated.”

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.



5.3 Database query

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-L05PRM4\SQLEXPRESS (SQL...)'. The 'airline_databases' database is expanded, showing 'dbo' tables. A red arrow points to 'dbo.Airline'. The main query window, titled 'SQLQuery5.sql - DESKTOP-L05PRM...', contains the following SQL query:

```
SELECT [Owndiscount], [AirlineName]
FROM [airline_databases].[dbo].[Airline]
```

The query is executed, and the Results pane shows the following data:

	Owndiscount	AirlineName
1	1	American Airlines
2	3	Delta Air Lines
3	22	Frontier Airlines
4	4	Hawaiian Airlines
5	10	Air China
6	3	Hainan Airlines
7	21	Shenzhen Airlines
8	12	Sichuan Airlines

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-L05PRM4\SQLEXPRESS (SQL...)'. The 'airline_databases' database is expanded, showing 'dbo' tables. A red arrow points to 'dbo.Airplane'. The main query window, titled 'SQLQuery6.sql - DESKTOP-L05PRM...', contains the following SQL query:

```
SELECT 1000 [AircraftNumber], [AircraftType]
FROM [airline_databases].[dbo].[Airplane]
```

The query is executed, and the Results pane shows the following data:

	AircraftNumber	AircraftType
1	1000	r-757
2	1000	y-247
3	1000	b-657
4	1000	b-727
5	1000	d-627
6	1000	y-757
7	1000	b-887
8	1000	b-888
9	1000	w-772
10	1000	b-727
11	1000	b-747

A red arrow points to the 'AircraftType' column in the results. A status bar at the bottom indicates 'Query executed successfully.'

Architecture assessment

*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.***

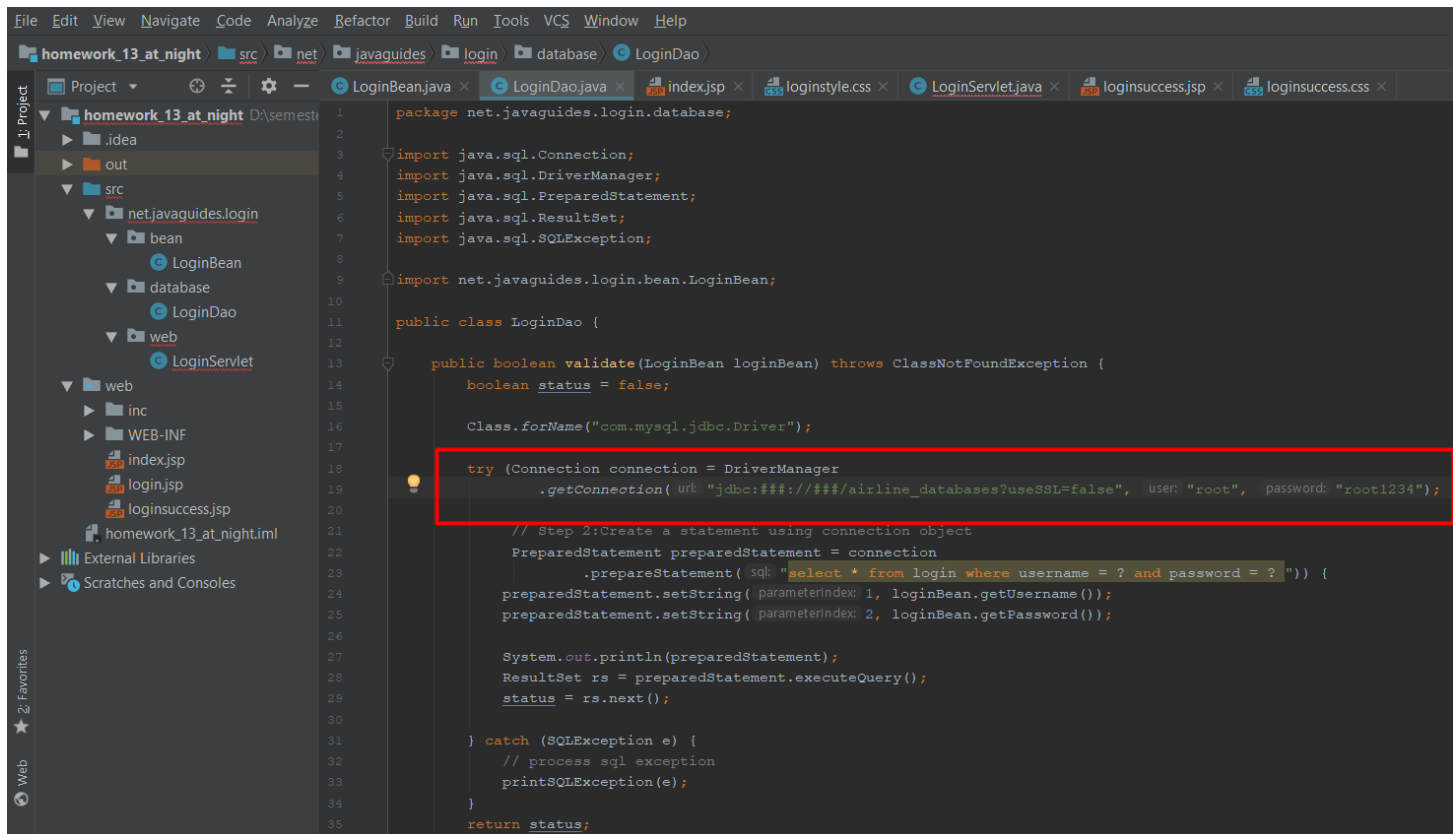
*If the two-tier C / S structure is selected, the server is only responsible for data management, and the logical processing, including flight comparison, will be carried out directly on the presentation page. That is to use java code to process data in JSP page, and then display the data on this page. 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.***

Program design description

6.1 Database connection

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



```
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", user: "root", password: "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         return status;
36     }
37 }
```

6.2 System menu

1. Customer opens the client page-----

- Select the city of departure
- destination
- Date of travel.
- Choose an airline.
- Price

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 (%).*

6.3 Data entry, modification and deletion

Data entry:

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-3UFN0U9\SQLEXPRESS (SQL Server 10.0.5008.8)'. The 'airlinedatabases' database is expanded, showing tables like 'dbo.Seat'. The SQL Query window on the right contains an INSERT statement for the 'dbo.Seat' table. The Messages pane at the bottom shows the execution results, indicating that the query was successful and affected rows.

SQLQuery1.sql - ...UFN0U9\HP (52)*

```
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');
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');
```

Messages

- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)
- (1 row(s) affected)

Query executed successfully.

DESKTOP-3UFN0U9\SQLEXPRESS ... DESKTOP-3UFN0U9\HP (52) airlinedatabases

Entry successful::

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'DESKTOP-L05PRM4\SQLEXPRESS (SQL Server 10.0.5008.8)'. The 'airline_database' is expanded, showing tables like 'dbo.Seat'. The SQL Query window on the right contains a SELECT statement for the 'dbo.Seat' table. The Results pane at the bottom shows the execution results, indicating that the query was successful and returned rows.

SQLQuery15.sql - DESKTOP-L05PRM4... SQLQuery14.sql - ...PRM4\oimdk (63)* SQLQuery13.sql - DESKTOP-L05PRM4\oimdk (63)*

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

NumberStatus	Type	Status
26	28	ECONOMY-CLASS 2
27	29	ECONOMY-CLASS 1
28	30	ECONOMY-CLASS 2
29	31	ECONOMY-CLASS 1
30	32	ECONOMY-CLASS 1
31	33	ECONOMY-CLASS 3
32	34	ECONOMY-CLASS 1
33	35	ECONOMY-CLASS 2
34	36	ECONOMY-CLASS 2
35	37	ECONOMY-CLASS 2
36	38	ECONOMY-CLASS 1
37	39	ECONOMY-CLASS 1
38	40	ECONOMY-CLASS 3
39	41	ECONOMY-CLASS 2
40	42	ECONOMY-CLASS 1
41	43	ECONOMY-CLASS 3
42	44	ECONOMY-CLASS 2
43	45	ECONOMY-CLASS 1
44	45	ECONOMY-CLASS 2

Query executed successfully.

File Edit View Query Debug Tools Window Community Help

New Query | Execute

airlinedatabases

Object Explorer

Connect >

DESKTOP-3UFN0U9\SQLEXPRESS (SQL Server 10.0.5512.1)

Databases

System Databases

Air booking management system

airlinedatabases

Database Diagrams

Tables

System Tables

dbo.Airline

dbo.Airplane

dbo.Airport

dbo.Anchor

dbo.Flight

dbo.Order

dbo.Passenger

dbo.Seat

dbo.useraccount

Views

Synonyms

Programmability

Service Broker

Storage

Security

library

myDatabase

store

teaching

xskc

Security

Server Objects

SQLQuery1.sql - ...UFN0U9\HP (52)*

```

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 ('434dd','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 ('34dd','b-774');
INSERT INTO dbo.Airplane (AircraftNumber,AircraftType) VALUES ('341dd','b-7447');
INSERT INTO dbo.Airplane (AircraftNumber,AircraftType) VALUES ('34dd','b-7744');
INSERT INTO dbo.Airplane (AircraftNumber,AircraftType) VALUES ('34dd','b-7733');
INSERT INTO dbo.Airplane (AircraftNumber,AircraftType) VALUES ('34dd','b-744');
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');

```

Messages

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

Query executed successfully.

DESKTOP-3UFN0U9\SQLEXPRESS ... DESKTOP-3UFN0U9\HP (52) airlinedatabases

Insert data again:

Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query | Execute

master

Object Explorer

Connect >

DESKTOP-L05PRM4\SQLEXPRESS (SQL Server 10.0.5512.1)

Databases

System Databases

airline_database

airline_databases

Database Diagrams

Tables

System Tables

dbo.Airline

dbo.Airplane

dbo.Airport

dbo.Anchor

dbo.Flight

dbo.Order

dbo.Passenger

dbo.Seat

dbo.user

dbo.useraccount

Views

Synonyms

Programmability

Service Broker

SQLQuery16.sql - DESKTOP-L05PRM4\oimdk (63)*

```

select [AircraftNumber]
,[AircraftType]
FROM [airline_databases].[dbo].[Airplane]

```

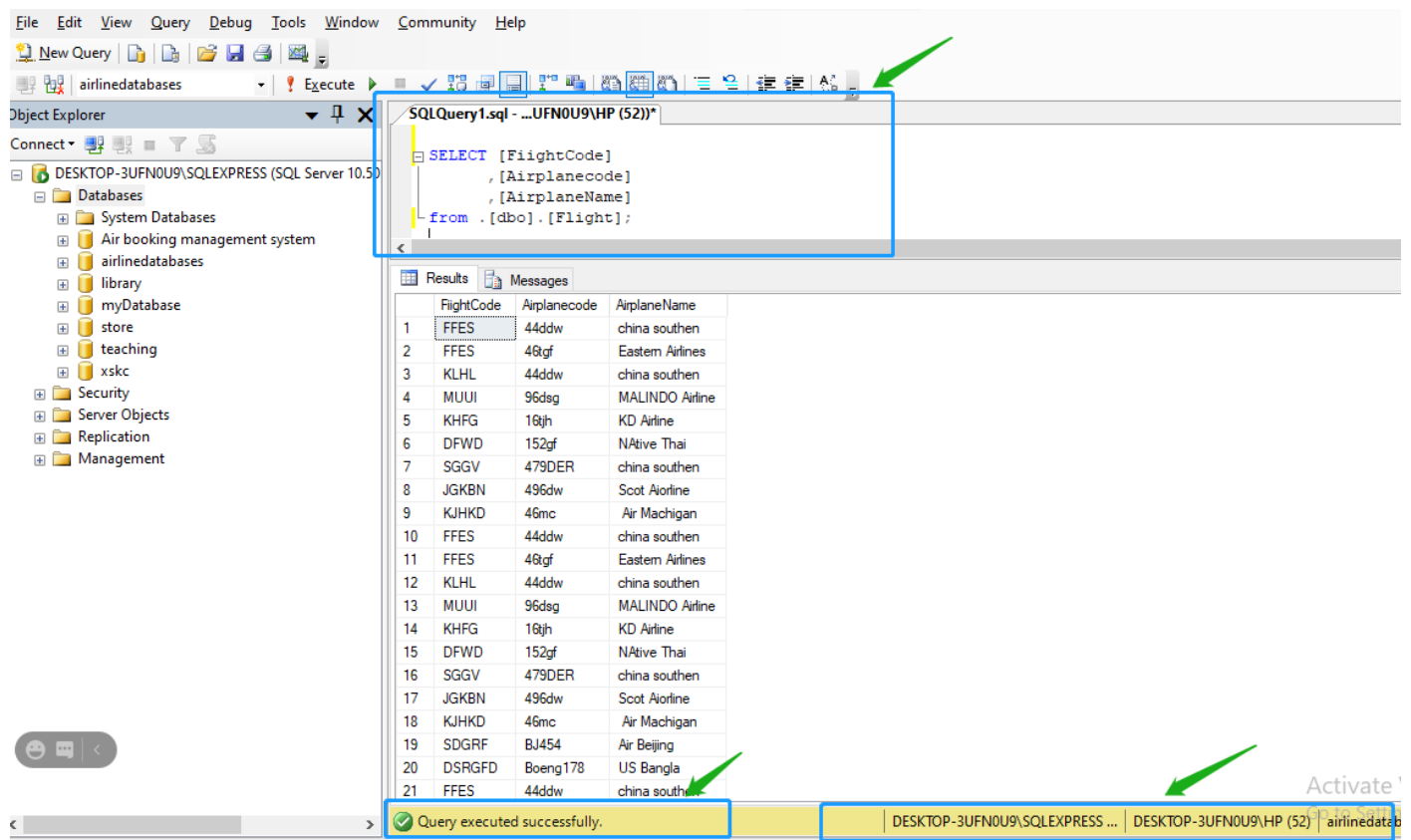
Results

	AircraftNumber	AircraftType
22	435dd	b-247
23	434dd	b-657
24	434dd	b-727
25	435dd	b-627
26	434dd	b-757
27	464dd	b-887
28	344dd	b-888
29	344dd	b-772
30	34d5d	b-727
31	434dd	b-747
32	434d4d	b-777
33	34dd	b-7744
34	341dd2	b-747
35	4314dd	b-747

Modification:

Airplane fare has been change by this query :

Update result:



SQLQuery1.sql - ...UFN0U9\HP (52)*

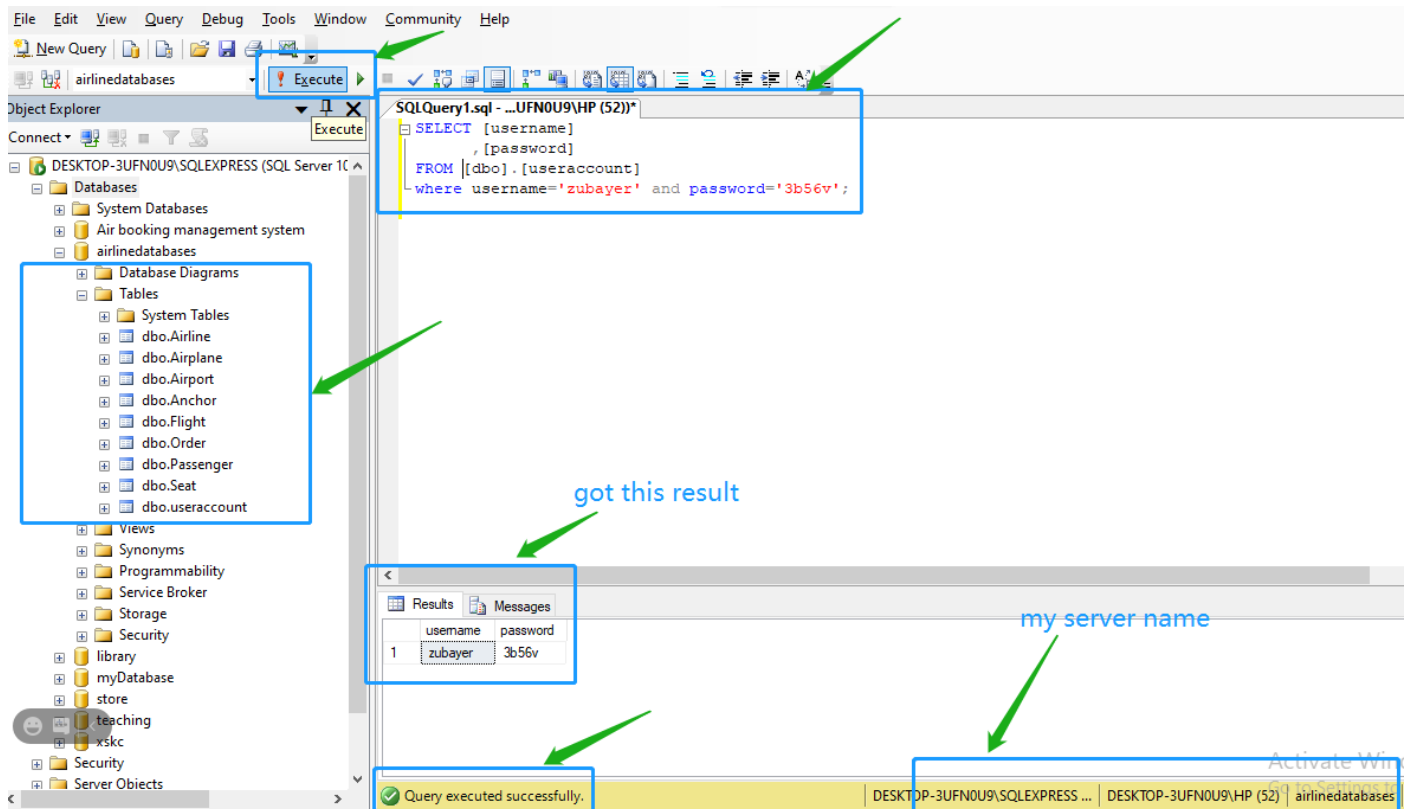
```
SELECT [FlightCode]
, [AirplaneCode]
, [AirplaneName]
from . [dbo] . [Flight];
```

	FlightCode	AirplaneCode	AirplaneName
1	FFES	44ddw	china southern
2	FFES	48gf	Eastern Airlines
3	KLHL	44ddw	china southern
4	MUUI	96dsg	MALINDO Airline
5	KHFG	18jh	KD Airline
6	DFWD	152gf	NAtive Thai
7	SGGV	479DER	china southern
8	JGKBN	496dw	Scot Aiorline
9	KJHKD	46mc	Air Machigan
10	FFES	44ddw	china southern
11	FFES	48gf	Eastern Airlines
12	KLHL	44ddw	china southern
13	MUUI	96dsg	MALINDO Airline
14	KHFG	18jh	KD Airline
15	DFWD	152gf	NAtive Thai
16	SGGV	479DER	china southern
17	JGKBN	496dw	Scot Aiorline
18	KJHKD	46mc	Air Machigan
19	SDGRF	BJ454	Air Beijing
20	DSRGFD	Boeng178	US Bangla
21	FFES	44ddw	china southern

Query executed successfully.

6.5 Database query

User login query



User login is successful

User fight-fare query or search

```
SELECT [DepatureAirport]
, [ArrivalAirport]
, [DepatureDate]
, [AirplaneName]
, [Fare]
FROM .[dbo].[Flight]
WHERE DepatureAirport='bejing'
AND ArrivalAirport='wuhan'
AND Depaturedate='2021-01-28'
```

```
OR AirplaneName='china southen'  
AND Fare<400;
```

The screenshot displays the SQL Server Enterprise Manager interface. The Object Explorer on the left shows the database structure, with a blue box highlighting the 'airlinedatabases' folder and its tables. The central query window shows the following SQL query:

```
SELECT [DepartureAirport]  
      , [ArrivalAirport]  
      , [DepartureDate]  
      , [AirplaneName]  
      , [Fare]  
FROM . [dbo] . [Flight]  
WHERE DepartureAirport='beijing'  
AND ArrivalAirport='wuhan'  
AND DepartureDate='2021-01-28'  
OR AirplaneName='china southen'  
AND Fare<400;
```

The query is executed, and the Results pane shows the following data:

	DepartureAirport	ArrivalAirport	DepartureDate	AirplaneName	Fare
1	beijing	wuhan	2021-01-28	china southen	342.3

Annotations in the image include:

- 'press to run' pointing to the Execute button in the toolbar.
- 'created tables' pointing to the 'airlinedatabases' folder in the Object Explorer.
- 'result' pointing to the Results pane.
- 'my server name' pointing to the server name 'DESKTOP-3UFN0U9\HP (52)' in the bottom status bar.

In here, a user search his/her requirement flight, this query in taking its own action and **find some flight details**.

Summary of Course Design

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.***

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.*

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/>

THANK YOU

苏贝尔-1811562127-ZUBAYER SM